|  |  |
| --- | --- |
| Университет ИТМО  Мега-факультет компьютерных технологий и управления  Факультет программная инженерия | |
| Дисциплина:  «» | |
| ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2  Вариант № 101143 | |
|  | Выполнил:  Проверил:  Кол-во баллов: |
| Санкт-Петербург  2021 г. | |

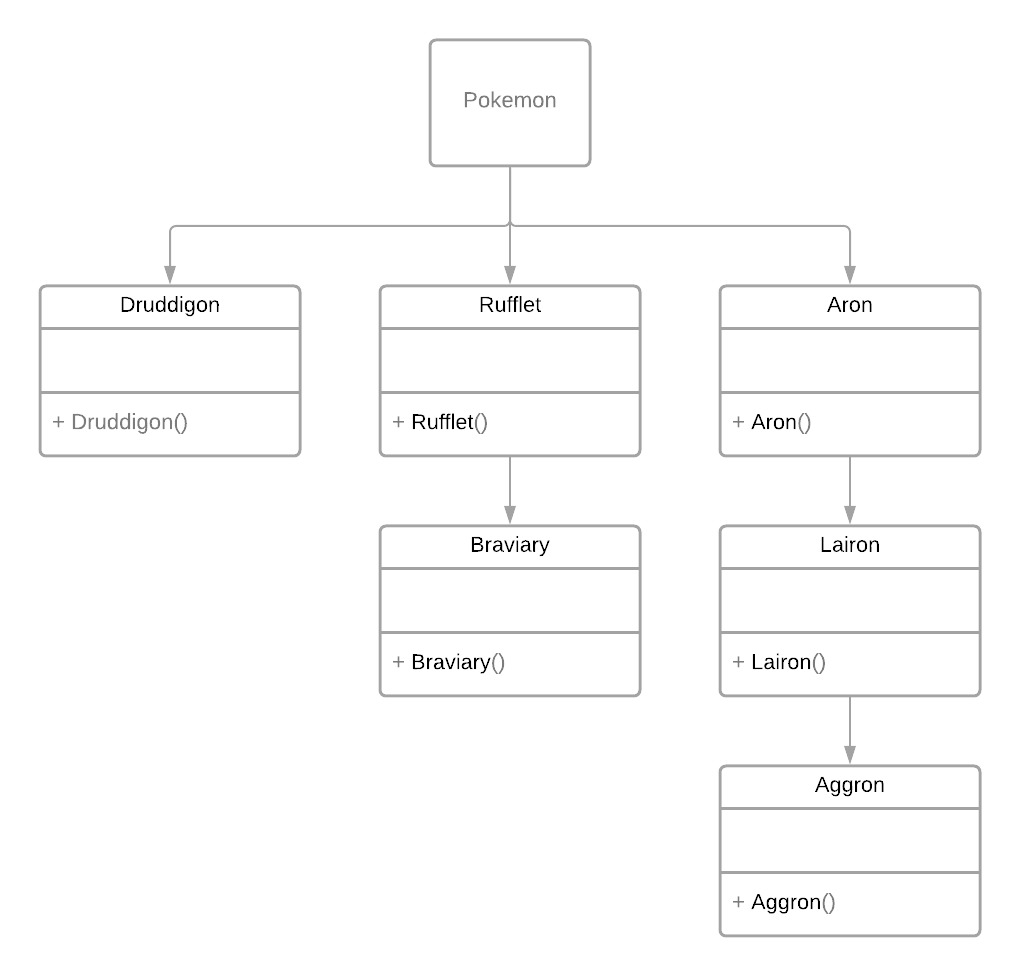
**Задание:**

1. Создать классы покемонов для своего варианта. Классы должны наследоваться в соответствии эволюции покемонов или от базового класса Pokemon . В конструкторе нужно задать типы покемонов и их базовые характеристики.
2. Создать классы атак для своего варианта. Классы должен наследоваться от классов PhysicalMove или SpecialMove или StatusMove. В конструкторе нужно будет задать тип атаки, ее силу и точность. После этого добавить атаки покемонам и проверить их действие в сражении. Переопределить метод describe, чтобы выводилось нужное сообщение.
3. Распределить покемонов по командам, запустить сражение.





**Иерархия классов покемонов:**



**Исходный код:**

**Git:** [**https://github.com/klim405/Labs/tree/main/JavaLabs/LabTwo**](https://github.com/klim405/Labs/tree/main/JavaLabs/LabTwo)

**LabTwo.class**

import ru.ifmo.se.pokemon.Battle;

import pokemons.Aron;

import pokemons.Druddigon;

import pokemons.Braviary;

import pokemons.Rufflet;

import pokemons.Lairon;

import pokemons.Aggron;

public class LabTwo {

// 101143

public static void main(String[] args) {

Battle b = new Battle();

b.addAlly(new Druddigon("Druddigon", (int) (Math.random()\*10)));

b.addAlly(new Aron("Aron", (int) (Math.random()\*10)));

b.addAlly(new Braviary("Braviary", (int) (Math.random()\*10)));

b.addFoe(new Rufflet("Rufflet", (int) (Math.random()\*10)));

b.addFoe(new Lairon("Lairon", (int) (Math.random()\*10)));

b.addFoe(new Aggron("Aggron", (int) (Math.random()\*10)));

b.go();

}

}

**moves/Confide.class**

package moves;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

import ru.ifmo.se.pokemon.StatusMove;

import ru.ifmo.se.pokemon.Stat;

public class Confide extends StatusMove {

public Confide() {

super(Type.NORMAL, 0.0D, 1.0D);

}

@Override

protected void applyOppEffects(Pokemon p) {

int currentStat = (int) p.getStat(Stat.SPECIAL\_ATTACK);

if (currentStat > -6) {

p.setMod(Stat.SPECIAL\_ATTACK, --currentStat);

}

}

@Override

protected boolean checkAccuracy(Pokemon pokemon, Pokemon pokemon1) {

return true;

}

@Override

protected String describe() {

return "использует Confide";

}

}

**moves/CrossPoison.class**

package moves;

import ru.ifmo.se.pokemon.PhysicalMove;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

import ru.ifmo.se.pokemon.Effect;

public class CrossPoison extends PhysicalMove {

public CrossPoison() {

super(Type.POISON, 70.0D, 1.0D);

}

@Override

protected void applyOppEffects(Pokemon p) {

if (Math.random() <= 0.1 && !(p.hasType(Type.STEEL) || p.hasType(Type.POISON))) {

Effect.poison(p);

}

}

@Override

protected double calcCriticalHit(Pokemon att, Pokemon def) {

if (Math.random() < 0.125) {

return 2.0D;

} else {

return 1.0D;

}

}

@Override

protected String describe() {

return "использует Cross Poison";

}

}

**moves/DoubleTeam.class**

package moves;

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 DoubleTeam extends StatusMove {

public DoubleTeam() {

super(Type.NORMAL, 0.0D, 1.0D);

}

@Override

protected void applySelfEffects(Pokemon p) {

int currentStat = (int) p.getStat(Stat.EVASION);

if (currentStat < 6) {

p.setMod(Stat.EVASION, ++currentStat);

}

}

@Override

protected boolean checkAccuracy(Pokemon pokemon, Pokemon pokemon1) {

return true;

}

@Override

protected String describe() {

return "использует Double Team";

}

}

**moves/** **Facade.class**

package moves;

import ru.ifmo.se.pokemon.PhysicalMove;

import ru.ifmo.se.pokemon.Type;

import ru.ifmo.se.pokemon.Status;

import ru.ifmo.se.pokemon.Pokemon;

public class Facade extends PhysicalMove {

public Facade() {

super(Type.NORMAL, 70.0D, 1.0D);

}

@Override

protected double calcBaseDamage(Pokemon att, Pokemon def) {

Status status = def.getCondition();

if (status == Status.BURN || status == Status.POISON || status == Status.PARALYZE) {

return super.calcBaseDamage(att, def) \* 2;

} else {

return super.calcBaseDamage(att, def);

}

}

@Override

protected String describe() {

return "использует Facade";

}

}

**moves/** **Flamethrower.class**

package moves;

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 Flamethrower extends SpecialMove {

public Flamethrower() {

super(Type.FIRE, 90.0D, 1.0D);

}

protected void applyOppEffects(Pokemon p) {

if (!p.hasType(Type.FIRE) && Math.random() <= 0.1) {

Effect.burn(p);

}

}

}

**moves/** **Harden.class**

package moves;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Stat;

import ru.ifmo.se.pokemon.Type;

import ru.ifmo.se.pokemon.StatusMove;

public class Harden extends StatusMove {

public Harden() {

super(Type.NORMAL, 0.0D, 1.0D);

}

@Override

protected void applySelfEffects(Pokemon p) {

int currentStat = (int) p.getStat(Stat.DEFENSE);

if (currentStat < 6) {

p.setMod(Stat.DEFENSE, ++currentStat);

}

}

@Override

protected boolean checkAccuracy(Pokemon pokemon, Pokemon pokemon1) {

return true;

}

@Override

protected String describe() {

return "использует Harden";

}

}

**moves/** **IceFang.class**

package moves;

import ru.ifmo.se.pokemon.Effect;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

import ru.ifmo.se.pokemon.PhysicalMove;

public class IceFang extends PhysicalMove {

public IceFang() {

super(Type.ICE, 65.0D, 0.95D);

}

@Override

protected void applyOppEffects(Pokemon p) {

if (Math.random() <= 0.1) {

Effect.flinch(p);

}

if (Math.random() <= 0.1) {

Effect.freeze(p);

}

}

@Override

protected String describe() {

return "использует Ice Fang";

}

}

**moves/** **Overheat.class**

package moves;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Stat;

import ru.ifmo.se.pokemon.Type;

import ru.ifmo.se.pokemon.SpecialMove;

public class Overheat extends SpecialMove {

public Overheat() {

super(Type.FIRE, 130.0D, 0.9D);

}

public void applyOppEffects(Pokemon p) {

int currentStat = (int) p.getStat(Stat.SPECIAL\_ATTACK);

if (currentStat >= -4) {

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

} else if (currentStat == -5) {

p.setMod(Stat.SPECIAL\_ATTACK, --currentStat);

}

}

@Override

protected String describe() {

return "использует Overheat";

}

}

**moves/** **Rest.class**

package moves;

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

public class Rest extends StatusMove {

public Rest() {

super(Type.PSYCHIC, 0.0D, 1.0D);

}

@Override

protected void applySelfEffects(Pokemon p) {

Effect e = new Effect().chance(0.0D).turns(2);

p.setMod(Stat.HP, (int) (p.getHP() - p.getStat(Stat.HP)));

p.addEffect(e);

}

@Override

protected boolean checkAccuracy(Pokemon pokemon, Pokemon pokemon1) {

return true;

}

@Override

protected String describe() {

return "использует Rest";

}

}

**moves/** **Screech.class**

package moves;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Stat;

import ru.ifmo.se.pokemon.Type;

import ru.ifmo.se.pokemon.StatusMove;

public class Screech extends StatusMove {

public Screech() {

super(Type.NORMAL, 0.0D, 0.85D);

}

@Override

protected void applyOppEffects(Pokemon p) {

int currentStat = (int) p.getStat(Stat.DEFENSE);

if (currentStat >= -4) {

p.setMod(Stat.DEFENSE, currentStat-2);

} else if (currentStat == -5) {

p.setMod(Stat.DEFENSE, --currentStat);

}

}

}

**moves/** **SmartStrike.class**

package moves;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

import ru.ifmo.se.pokemon.PhysicalMove;

public class SmartStrike extends PhysicalMove {

public SmartStrike() {

super(Type.STEEL, 70.0D, 1.0D);

}

@Override

protected boolean checkAccuracy(Pokemon pokemon, Pokemon pokemon1) {

return true;

}

@Override

protected String describe() {

return "использует Smart Strike";

}

}

**pokemons/** **Aggron.class**

package pokemons;

import moves.SmartStrike;

public class Aggron extends Lairon {

public Aggron(String name, int level) {

super(name, level);

setStats(70, 110, 180, 60, 60, 50);

addMove(new SmartStrike());

}

}

**pokemons/** **Aron.class**

package pokemons;

import ru.ifmo.se.pokemon.Pokemon;

import moves.DoubleTeam;

import moves.Flamethrower;

import ru.ifmo.se.pokemon.Type;

public class Aron extends Pokemon {

public Aron(String name, int level) {

super(name, level);

setStats(50, 70, 100, 40, 40, 30);

setType(Type.STEEL, Type.ROCK);

addMove(new Flamethrower());

addMove(new DoubleTeam());

}

}

**pokemons/** **Braviary.class**

package pokemons;

import moves.CrossPoison;

public class Braviary extends Rufflet{

public Braviary(String name, int level) {

super(name, level);

setStats(100, 123, 75, 57, 75, 80);

addMove(new CrossPoison());

}

}

**pokemons/** **Druddigon.class**

package pokemons;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

import moves.Facade;

import moves.Harden;

import moves.IceFang;

import moves.Rest;

public class Druddigon extends Pokemon {

public Druddigon(String name, int level) {

super(name, level);

setStats(77, 120, 90, 60, 90, 48);

setType(Type.DRAGON);

addMove(new Facade());

addMove(new Harden());

addMove(new Rest());

addMove(new IceFang());

}

}

**pokemons/** **Lairon.class**

package pokemons;

import moves.Overheat;

public class Lairon extends Aron {

public Lairon(String name, int level) {

super(name, level);

setStats(60, 90, 140, 50, 50, 40);

addMove(new Overheat());

}

}

**pokemons/** **Rufflet.class**

package pokemons;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

import moves.Confide;

import moves.Facade;

import moves.Screech;

public class Rufflet extends Pokemon {

public Rufflet(String name, int level) {

super(name, level);

setStats(70, 83, 50, 37, 50, 60);

setType(Type.NORMAL, Type.FLYING);

addMove(new Confide());

addMove(new Facade());

addMove(new Screech());

}

}

**Вывод:**

Я научился работать с классами Java. Это пригодится для абстрактного описания объектов.