**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

Факультет Программной Инженерии и Компьютерной Техники

Дисциплина: «Программирование»

**ОТЧЁТ**

по лабораторной работе №2  
Вариант №2636

Выполнил:

Студент группы P3108

Грищенко Андрей Викторович

Проверила:

Харитонова Анастасия Евгеньевна

Санкт-Петербург

2023 г.

# 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)

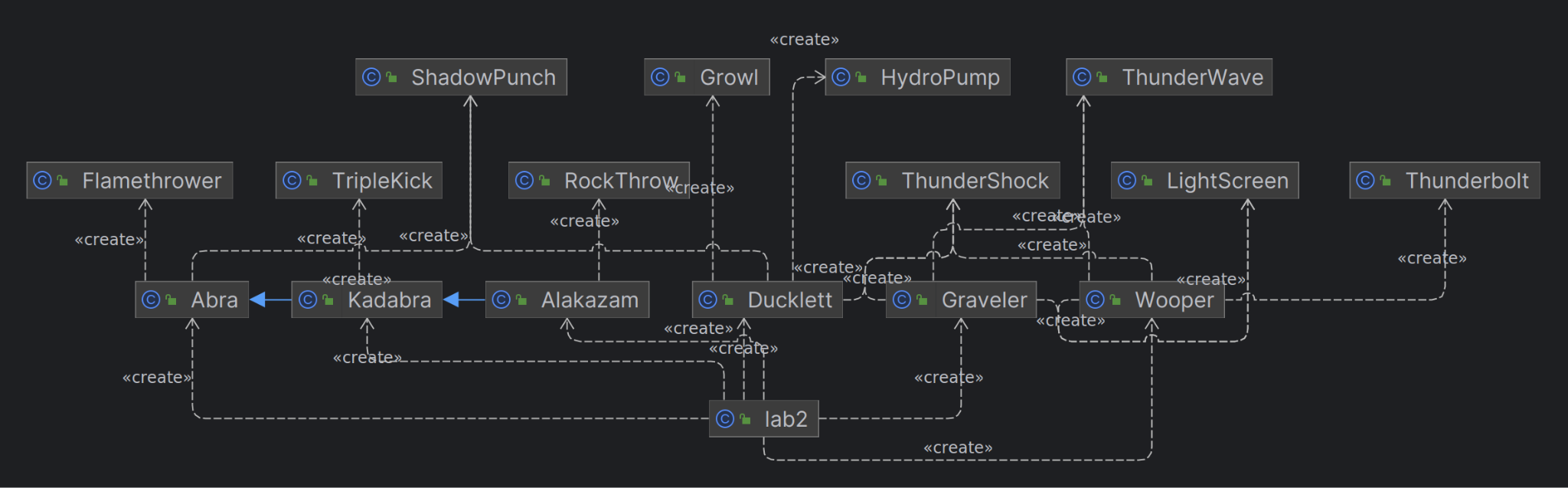
Классы покемонов должны наследоваться в соответствии с цепочкой эволюции покемонов. На основе базовых классов PhysicalMove, SpecialMove и StatusMove реализовать свои классы для заданных видов атак.

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.



# UML-диаграмма классов:



*Рисунок 1*

# Код программы:

# Покемоны:

package *P*okemons;

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

import PhysicalMoves.\*;

import SpecialMoves.\*;

public class Abra extends Pokemon{

public Abra(String name, int level) {

super(name, level);

setType(Type.PSYCHIC);

setStats(2, 2, 1, 7, 4, 6);

setMove(new Flamethrower(), new ShadowPunch());

}

}

package *P*okemons;

import PhysicalMoves.\*;

public class Alakazam extends Kadabra {

public Alakazam(String name, int level) {

super(name, level);

setStats(4, 3, 3, 8, 6, 8);

addMove(new RockThrow());

}

}

package *P*okemons;

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

import PhysicalMoves.\*;

import SpecialMoves.\*;

import StatusMoves.\*;

public class Ducklett extends Pokemon {

public Ducklett(String name, int level) {

super(name, level);

setType(Type.WATER, Type.FLYING);

setStats(4, 3, 3, 3, 3, 4);

setMove(new Growl(), new ThunderShock(), new HydroPump(), new ShadowPunch());

}

}

package *P*okemons;

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

import SpecialMoves.\*;

import StatusMoves.\*;

public class Graveler extends Pokemon {

public Graveler(String name, int level) {

super(name, level);

setType(Type.ROCK, Type.GROUND);

setStats(4, 6, 7, 3, 3, 3);

setMove(new LightScreen(), new ThunderShock(), new ThunderWave());

}

}

package *P*okemons;

import PhysicalMoves.\*;

public class Kadabra extends Abra {

public Kadabra(String name, int level) {

super(name, level);

setStats(3, 3, 2, 8, 5, 7);

addMove(new TripleKick());

}

}

package *P*okemons;

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

import SpecialMoves.\*;

import StatusMoves.\*;

public class Wooper extends Pokemon {

public Wooper(String name, int level) {

super(name, level);

setType(Type.WATER, Type.GROUND);

setStats(4, 3, 3, 2, 2, 1);

setMove(new LightScreen(), new ThunderShock(), new ThunderWave(), new Thunderbolt());

}

}

# Атаки:

package *P*hysical*M*oves;

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

public class RockThrow extends PhysicalMove {

public RockThrow() {

super(Type.ROCK, 50.0, 90.0);

}

public String describe() {

return "using physical attack Rock Throw";

}

}

package *P*hysical*M*oves;

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

public class ShadowPunch extends PhysicalMove {

public ShadowPunch() {

super(Type.GHOST, 60.0, 0.0);

}

public boolean checkAccuracy(Pokemon att, Pokemon def) {

return true;

}

public String describe() {

return "using physical attack Shadow Punch";

}

}

package *P*hysical*M*oves;

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

public class TripleKick extends PhysicalMove {

public TripleKick() {

super(Type.FIGHTING, 10.0, 90.0);

}

public String describe() {

return "using physical attack Triple Kick";

}

public double calcBaseDamage(Pokemon attacker, Pokemon defender) {

double damage = super.calcBaseDamage(attacker, defender);

this.power = 20;

damage += super.calcBaseDamage(attacker, defender);

this.power = 30;

damage += super.calcBaseDamage(attacker, defender);

return damage;

}

}

package *S*pecial*M*oves;

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

public class Flamethrower extends SpecialMove {

public Flamethrower() {

super(Type.FIRE, 90.0, 100.0);

}

public void applyOppEffects(Pokemon p) {

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

Effect.burn(p);

}

}

public String describe() {

return "using special attack Flamethrower";

}

}

package *S*pecial*M*oves;

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

public class HydroPump extends SpecialMove {

public HydroPump() {

super(Type.WATER, 110.0, 80.0);

}

public String describe() {

return "using special attack Hydro Pump";

}

}

package *S*pecial*M*oves;

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

public class Thunderbolt extends SpecialMove {

public Thunderbolt() {

super(Type.ELECTRIC, 90.0, 100.0);

}

public void applyOppEffects(Pokemon p) {

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

Effect.paralyze(p);

}

}

public String describe() {

return "using special attack Thunderbolt";

}

}

package *S*pecial*M*oves;

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

public class ThunderShock extends SpecialMove{

public ThunderShock() {

super(Type.ELECTRIC, 40.0, 100.0);

}

public void applyOppEffects(Pokemon p) {

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

Effect.paralyze(p);

}

}

public String describe() {

return "using special attack Thunder Shock";

}

}

package *S*tatus*M*oves;

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

public class Growl extends StatusMove {

public Growl() {

super(Type.NORMAL, 0.0, 100.0);

}

public void applyOppEffects(Pokemon p) {

p.setMod(Stat.ATTACK, -1);

}

public String describe() {

return "using status attack Growl";

}

}

package *S*tatus*M*oves;

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

public class LightScreen extends StatusMove {

public LightScreen() {

super(Type.PSYCHIC, 0.0, 0.0);

}

public void applySelfEffects(Pokemon p) {

p.setMod(Stat.HP, 1);

}

public boolean checkAccuracy(Pokemon att, Pokemon def) {

return true;

}

public String describe() {

return "using status attack Light Screen";

}

}

package *S*tatus*M*oves;

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

public class ThunderWave extends StatusMove {

public ThunderWave() {

super(Type.ELECTRIC, 0.0, 90.0);

}

public void applyOppEffects(Pokemon p) {

if (!p.hasType(Type.ELECTRIC)) {

Effect.paralyze(p);

}

}

public String describe() {

return "using status attack Thunder Wave";

}

}

# Главный класс (lab2.java):

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

import Pokemons.\*;

public class lab2 {

public static void main(String[] args) {

Battle b = new Battle();

Pokemon ducklett = new Ducklett("Ducklett1", 1);

Pokemon graveler = new Graveler("Graveler1", 1);

Pokemon wooper = new Wooper("Wooper1", 1);

Pokemon abra = new Abra("Abra2", 1);

Pokemon kadabra = new Kadabra("Kadabra2", 1);

Pokemon alakazam = new Alakazam("Alakazam2", 1);

b.addAlly(ducklett);

b.addAlly(graveler);

b.addAlly(wooper);

b.addFoe(abra);

b.addFoe(kadabra);

b.addFoe(alakazam);

b.go();

}

}

# Результат работы программы:

Ducklett Ducklett1 from the team Red enters the battle!

Abra Abra2 from the team Blue enters the battle!

Abra Abra2 using physical attack Shadow Punch.

Ducklett Ducklett1 loses 5 hit points.

Ducklett Ducklett1 using physical attack Shadow Punch.

Abra Abra2 loses 9 hit points.

Abra Abra2 using physical attack Shadow Punch.

Ducklett Ducklett1 loses 5 hit points.

Ducklett Ducklett1 using physical attack Shadow Punch.

Abra Abra2 loses 11 hit points.

Abra Abra2 faints.

Kadabra Kadabra2 from the team Blue enters the battle!

Kadabra Kadabra2 using special attack Flamethrower.

Ducklett Ducklett1 loses 3 hit points.

Ducklett Ducklett1 faints.

Graveler Graveler1 from the team Red enters the battle!

Kadabra Kadabra2 using physical attack Shadow Punch.

Graveler Graveler1 loses 4 hit points.

Graveler Graveler1 using special attack Thunder Shock.

Kadabra Kadabra2 loses 4 hit points.

Kadabra Kadabra2 is paralyzed

Kadabra Kadabra2 using physical attack Shadow Punch.

Critical hit!

Graveler Graveler1 loses 11 hit points.

Graveler Graveler1 faints.

Wooper Wooper1 from the team Red enters the battle!

Kadabra Kadabra2 using special attack Flamethrower.

Critical hit!

Wooper Wooper1 loses 6 hit points.

Wooper Wooper1 using status attack Light Screen.

Wooper Wooper1 loses 1 hit points.

Kadabra Kadabra2 using special attack Flamethrower.

Critical hit!

Wooper Wooper1 loses 6 hit points.

Wooper Wooper1 faints.

Team Red loses its last Pokemon.

The team Blue wins the battle!

# Заключение

В результате проделанной лабораторной работы я изучил концепцию ООП и методы ее реализации в языке программирования Java, а также работу с внешними jar пакетами.