Министерство образования и науки РФ

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 №5215

*Выполнил:*

Студент группы P3109

Mallaev Sabur N.

*Преподаватель:*

Шешуков Дмитрий

Михайлович



Санкт-Петербург, 2021

Цель

Цель работы: на простом примере разобраться с основными концепциями ООП и научиться использовать их в программах.

Задание

* Ознакомиться с [документацией](https://se.ifmo.ru/~tony/doc/), обращая особое внимание на классы Pokemon и Move. При дальнейшем выполнении лабораторной работы читать документацию еще несколько раз.
* Скачать файл Pokemon.jar. Его необходимо будет использовать как для компиляции, так и для запуска программы. Распаковывать его не надо! Нужно научиться подключать внешние jar-файлы к своей программе.
* Написать минимально работающую программу и посмотреть как она работает.

Battle b = new Battle();

Pokemon p1 = new Pokemon("Чужой", 1);

Pokemon p2 = new Pokemon("Хищник", 1);

b.addAlly(p1);

b.addFoe(p2);

b.go();

* Создать один из классов покемонов для своего варианта. Класс должен наследоваться от базового класса Pokemon. В конструкторе нужно будет задать типы покемона и его базовые характеристики. После этого попробуйте добавить покемона в сражение.
* Создать один из классов атак для своего варианта (лучше всего начать с физической или специальной атаки). Класс должен наследоваться от класса PhysicalMove или SpecialMove. В конструкторе нужно будет задать тип атаки, ее силу и точность. После этого добавить атаку покемону и проверить ее действие в сражении. Не забудьте переопределить метод describe, чтобы выводилось нужное сообщение.
* Если действие атаки отличается от стандартного, например, покемон не промахивается, либо атакующий покемон также получает повреждение, то в классе атаки нужно дополнительно переопределить соответствующие методы (см. документацию). При реализации атак, которые меняют статус покемона (наследники StatusMove), скорее всего придется разобраться с классом Effect. Он позволяет на один или несколько ходов изменить состояние покемона или модификатор его базовых характеристик.
* Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.



Диаграмма классов реализованной объектной модели

Ад.

Исходный код программы

Lab1.java

package info;

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

public class Lab1 {

public static void main(String[] args) {

Battle b = new Battle();

b.addAlly(new Sawk("Kwas", 1));

b.addAlly(new Binacle("Elcanib", 1));

b.addAlly(new Barbaracle("barbarthing", 1));

b.addFoe(new Cleffa("Lame Clef", 1));

b.addFoe(new Clefairy("Normal Clef", 1));

b.addFoe(new Clefable("Cool Clef", 1));

b.go();

}

}

Pokemon

Sawk.java

package info;

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

public class Sawk extends Pokemon {

public Sawk(String name, int level) {

super(name, level);

this.setType(Type.FIGHTING);

this.setStats(75, 125, 75, 30, 75, 85);

this.setMove(new Payback(),

new RockTomb(),

new BulkUp(),

new ShiftGear());

}

public Sawk() {

this("Безымянный", 1);

}

}

Binacle.java

package info;

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

public class Binacle extends Pokemon {

public Binacle(String name, int level) {

super(name, level);

this.setType(Type.ROCK, Type.WATER);

this.setStats(42, 52, 67, 39, 56, 50);

this.setMove(new SpacialRend(),

new Headbutt(),

new Swagger());

}

public Binacle() {

this("Безымянный", 1);

}

}

Barbaracle.java

package info;

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

public class Barbaracle extends Binacle {

public Barbaracle(String name, int level) {

super(name, level);

this.setStats(72, 105, 115, 54, 86, 68);

this.setMove(new SpacialRend(),

new Headbutt(),

new Swagger(),

new BulkUp());

}

public Barbaracle() {

this("Безымянный", 1);

}

}

Cleffa.java

package info;

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

public class Cleffa extends Pokemon {

public Cleffa(String name, int level) {

super(name, level);

this.setType(Type.FAIRY);

this.setStats(50, 25, 28, 45, 55, 15);

this.setMove(new Quash(), new FireBlast());

}

public Cleffa() {

this("Безымянный", 1);

}

}

Clefairy.java

package info;

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

public class Clefairy extends Cleffa {

public Clefairy(String name, int level) {

super(name, level);

this.setStats(70, 45, 48, 60, 65, 35);

this.setMove(new Quash(), new FireBlast(), new Pound());

}

public Clefairy() {

this("Безымянный", 1);

}

}

Clefable.java

package info;

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

public class Clefable extends Clefairy {

public Clefable(String name, int level) {

super(name, level);

this.setStats(95, 70, 73, 95, 90, 60);

this.setMove(new Quash(), new FireBlast(), new Pound(), new Swagger());

}

public Clefable() {

this("Безымянный", 1);

}

}

Moves

BulkUp.java

package info;

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

public class BulkUp extends StatusMove {

public BulkUp() {

super(Type.FIGHTING, 0, 0);

}

public String describe() {

return "uses Bulk Up";

}

public boolean checkAccuracy(Pokemon att, Pokemon def) {

return true;

}

public void applySelfEffects(Pokemon p) {

p.setMod(Stat.ATTACK, 2);

p.setMod(Stat.DEFENSE, 2);

}

}

FireBlast.java

package info;

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

public class FireBlast extends SpecialMove {

public FireBlast() {

super(Type.FIRE, 110, 0.85);

}

public String describe() {

return "uses Fire Blast";

}

public void applyOppEffects(Pokemon pokemon) {

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

Effect.burn(pokemon);

}

}

}

Headbutt.java

package info;

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

public class Headbutt extends PhysicalMove {

public Headbutt() {

super(Type.NORMAL, 70, 1.00);

}

public String describe() {

return "uses Headbutt";

}

public void applyOppEffects(Pokemon p) {

if (Math.random() <= 0.3) {

Effect.flinch(p);

}

}

}

Payback.java

package info;

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

public class Payback extends PhysicalMove {

public Payback() {

super(Type.DARK, 50, 1.00);

}

public String describe() {

return "uses Payback";

}

//The actual effect of payback cannot be made

//with this system because i have no clue if pokemon

//goes first or second.

//Improve your documentation, it's really bad!

}

Pound.java

package info;

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

public class Pound extends PhysicalMove {

public Pound() {

super(Type.NORMAL, 40, 1.00);

}

public String describe() {

return "uses Pound";

}

}

Quash.java

package info;

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

public class Quash extends StatusMove {

public Quash() {

super(Type.DARK, 0, 1.00, 1, 1);

}

public String describe() {

return "wastes its turn because Quash is a useless move";

}

}

RockTomb.java

package info;

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

public class RockTomb extends PhysicalMove {

public RockTomb() {

super(Type.ROCK, 60, 0.95);

}

public String describe() {

return "uses Rock Tomb";

}

public void applyOppEffects(Pokemon p) {

p.setMod(Stat.SPEED, -2);

}

}

ShiftGear.java

package info;

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

public class ShiftGear extends StatusMove {

public ShiftGear() {

super(Type.FIGHTING, 0, 0);

}

public String describe() {

return "uses Shift Gear";

}

public boolean checkAccuracy(Pokemon att, Pokemon def) {

return true;

}

public void applySelfEffects(Pokemon p) {

p.setMod(Stat.ATTACK, 2);

p.setMod(Stat.SPEED, 3);

}

}

SpacialRend.java

package info;

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

public class SpacialRend extends SpecialMove {

public SpacialRend() {

super(Type.DRAGON, 100, 0.95);

}

public String describe() {

return "uses Spacial Rend";

}

//The entire point of this move is increased critical hit ratio

//but it looks like this system doesn't even have crits

}

Swagger.java

package info;

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

public class Swagger extends StatusMove {

public Swagger() {

super(Type.NORMAL, 0, 0.85);

}

public String describe() {

return "used Swagger";

}

public boolean checkAccuracy(Pokemon att, Pokemon def) {

return true;

}

public void applyOppEffects(Pokemon pokemon) {

pokemon.setMod(Stat.ATTACK, 2);

Effect.confuse(pokemon);

}

}

Результат работы программы

На самом деле результатов может быть гораздо больше из-за элемента случайности, но вот одна попытка, которая не слилась минутами, как некоторые другие.

Sawk Kwas from the team Yellow enters the battle!

Cleffa Lame Clef from the team Blue enters the battle!

Cleffa Lame Clef wastes its turn because Quash is a useless move.

Sawk Kwas uses Rock Tomb.

Cleffa Lame Clef loses 5 hit points.

Cleffa Lame Clef decreases speed.

Cleffa Lame Clef misses

Sawk Kwas uses Rock Tomb.

Cleffa Lame Clef loses 4 hit points.

Cleffa Lame Clef decreases speed.

Sawk Kwas uses Bulk Up.

Sawk Kwas increases attack.

Sawk Kwas increases defense.

Cleffa Lame Clef uses Fire Blast.

Sawk Kwas loses 7 hit points.

Cleffa Lame Clef wastes its turn because Quash is a useless move.

Sawk Kwas uses Rock Tomb.

Cleffa Lame Clef loses 5 hit points.

Cleffa Lame Clef decreases speed.

Cleffa Lame Clef faints.

Clefairy Normal Clef from the team Blue enters the battle!

Sawk Kwas uses Shift Gear.

Sawk Kwas increases attack.

Sawk Kwas increases speed.

Clefairy Normal Clef uses Fire Blast.

Sawk Kwas loses 4 hit points.

Sawk Kwas uses Bulk Up.

Sawk Kwas increases attack.

Sawk Kwas increases defense.

Clefairy Normal Clef uses Fire Blast.

Sawk Kwas loses 7 hit points.

Sawk Kwas faints.

Binacle Elcanib from the team Yellow enters the battle!

Clefairy Normal Clef wastes its turn because Quash is a useless move.

Binacle Elcanib uses Spacial Rend.

Clefairy Normal Clef loses 1 hit points.

Clefairy Normal Clef isn't affected by DRAGON

Binacle Elcanib uses Spacial Rend.

Clefairy Normal Clef loses 1 hit points.

Clefairy Normal Clef isn't affected by DRAGON

Clefairy Normal Clef uses Fire Blast.

Binacle Elcanib loses 1 hit points.

Binacle Elcanib used Swagger.

Clefairy Normal Clef increases attack.

Clefairy Normal Clef uses Pound.

Binacle Elcanib loses 2 hit points.

Binacle Elcanib used Swagger.

Clefairy Normal Clef increases attack.

Clefairy Normal Clef hits himself in confusion.

Clefairy Normal Clef loses 4 hit points.

Binacle Elcanib uses Headbutt.

Clefairy Normal Clef loses 4 hit points.

Clefairy Normal Clef uses Fire Blast.

Binacle Elcanib loses 2 hit points.

Binacle Elcanib uses Headbutt.

Clefairy Normal Clef loses 3 hit points.

Clefairy Normal Clef faints.

Clefable Cool Clef from the team Blue enters the battle!

Clefable Cool Clef uses Fire Blast.

Binacle Elcanib loses 2 hit points.

Binacle Elcanib uses Spacial Rend.

Clefable Cool Clef loses 1 hit points.

Clefable Cool Clef isn't affected by DRAGON

Clefable Cool Clef uses Pound.

Binacle Elcanib loses 2 hit points.

Binacle Elcanib uses Spacial Rend.

Clefable Cool Clef loses 1 hit points.

Clefable Cool Clef isn't affected by DRAGON

Clefable Cool Clef uses Fire Blast.

Binacle Elcanib loses 2 hit points.

Binacle Elcanib uses Spacial Rend.

Clefable Cool Clef loses 1 hit points.

Clefable Cool Clef isn't affected by DRAGON

Clefable Cool Clef wastes its turn because Quash is a useless move.

Binacle Elcanib used Swagger.

Clefable Cool Clef increases attack.

Clefable Cool Clef hits himself in confusion.

Clefable Cool Clef loses 5 hit points.

Binacle Elcanib used Swagger.

Clefable Cool Clef increases attack.

Clefable Cool Clef hits himself in confusion.

Clefable Cool Clef loses 4 hit points.

Binacle Elcanib uses Headbutt.

Clefable Cool Clef loses 3 hit points.

Clefable Cool Clef faints.

Team Blue loses its last Pokemon.

**The team Yellow wins the battle!**

Выводы

Объектно-ориентированное программирование – это очень удобно, однако, пример “Каждая эволюция одного покемона – это еще один наследник этого класса”, как мне кажется, не совсем удачный.