Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Программирование»

По лабораторной работе №2

Основные концепции ООП

Вариант 2650

Студент

*Александров А. А.*

*Р3106*

Преподаватель

*Письмак А.Е.*

Санкт-Петербург 2023

Содержание

[Задание 4](#_Toc149565835)

[Покемоны 5](#_Toc149565836)

[UML диаграмма реализованных классов 5](#_Toc149565837)

[Код программы 6](#_Toc149565838)

[Main.java 6](#_Toc149565839)

[Pokemons.info 6](#_Toc149565840)

[Magearna.java 6](#_Toc149565841)

[Grimer.java 6](#_Toc149565842)

[Muk.java 6](#_Toc149565843)

[Magearna.java 7](#_Toc149565844)

[Bounsweet.java 7](#_Toc149565845)

[Steenee.java 7](#_Toc149565846)

[Tsareena.java 7](#_Toc149565847)

[Abilities 8](#_Toc149565848)

[Confide.java 8](#_Toc149565849)

[DarkPulse.java 8](#_Toc149565850)

[DoubleTeam.java 9](#_Toc149565851)

[FlashCannon.java 9](#_Toc149565852)

[MetalClaw.java 9](#_Toc149565853)

[NightSlash.java 10](#_Toc149565854)

[PlayNice.java 10](#_Toc149565855)

[RazorLeaf.java 11](#_Toc149565856)

[SludgeBomb.java 11](#_Toc149565857)

[Splash.java 12](#_Toc149565858)

[Swagger.java 12](#_Toc149565859)

[Результат выполнения программы 13](#_Toc149565860)

[Заключение 15](#_Toc149565861)

# 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.

Базовые классы и симулятор сражения находятся в [jar-архиве](https://se.ifmo.ru/documents/10180/660917/Pokemon.jar/a7ce60af-6ee6-47d0-a95e-e5ed9a697bd2) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](https://se.ifmo.ru/~tony/doc/).

Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах [http://poke-universe.ru](http://poke-universe.ru/), [http://pokemondb.net](http://pokemondb.net/),<http://veekun.com/dex/pokemon>

# Изображение выглядит как текст, мультфильм, иллюстрация Автоматически созданное описаниеИзображение выглядит как текст, иллюстрация Автоматически созданное описаниеПокемоны

# Изображение выглядит как снимок экрана, черный, линия, прямоугольный Автоматически созданное описаниеUML диаграмма реализованных классов

# Код программы

## Main.java

import ru.ifmo.se.pokemon.\*;  
import pokemons.info.\*;  
  
public class Main {  
 public static void main(String[] args) {  
 Battle b = new Battle();  
 Magearna p1 = new Magearna("RoboGirl", 1);  
 Grimer p2 = new Grimer("Kucha", 1);  
 Muk p3 = new Muk("BigKucha", 1);  
 Bounsweet p4 = new Bounsweet("LittleBerry", 1);  
 Steenee p5 = new Steenee("MediumBerry", 1);  
 Tsareena p6 = new Tsareena("BigBerry", 1);  
 b.addAlly(p1);  
 b.addAlly(p2);  
 b.addAlly(p3);  
 b.addFoe(p4);  
 b.addFoe(p5);  
 b.addFoe(p6);  
 b.go();  
 }  
}

## Pokemons.info

### Magearna.java

package pokemons.info;  
import ru.ifmo.se.pokemon.\*;  
import abilities.\*;  
  
public class Magearna extends Pokemon{  
 public Magearna(String Name, int Lvl){  
 super(Name, Lvl);  
 setStats(80, 95, 115, 130, 115, 65);  
 setType(Type.*STEEL*, Type.*FAIRY*);  
 setMove(new FlashCannon(), new Confide(), new MetalClaw(), new NightSlash());  
 }  
}

### Grimer.java

package pokemons.info;  
import ru.ifmo.se.pokemon.\*;  
import abilities.\*;  
  
public class Grimer extends Pokemon{  
 public Grimer(String Name, int Lvl){  
 super(Name, Lvl);  
 setStats(80, 80, 50, 40, 50, 25);  
 setType(Type.*POISON*);  
 setMove(new SludgeBomb(), new DoubleTeam(), new Swagger());  
 }  
}

### Muk.java

package pokemons.info;  
import ru.ifmo.se.pokemon.\*;  
import abilities.\*;  
  
public class Muk extends Grimer {  
 public Muk(String Name, int Lvl){  
 super(Name, Lvl);  
 setStats(105, 105, 75, 65, 100, 50);  
 setType(Type.*POISON*);  
 setMove(new SludgeBomb(), new DoubleTeam(), new Swagger(), new DarkPulse());  
 }  
}

### Magearna.java

package pokemons.info;  
import ru.ifmo.se.pokemon.\*;  
import abilities.\*;  
  
public class Magearna extends Pokemon{  
 public Magearna(String Name, int Lvl){  
 super(Name, Lvl);  
 setStats(80, 95, 115, 130, 115, 65);  
 setType(Type.*STEEL*, Type.*FAIRY*);  
 setMove(new FlashCannon(), new Confide(), new MetalClaw(), new NightSlash());  
 }  
}

### Bounsweet.java

package pokemons.info;  
import ru.ifmo.se.pokemon.\*;  
import abilities.\*;  
  
public class Bounsweet extends Pokemon{  
 public Bounsweet(String Name, int Lvl){  
 super(Name, Lvl);  
 setStats(42, 30, 38, 30, 38, 32);  
 setType(Type.*GRASS*);  
 setMove(new RazorLeaf(), new Splash());  
 }  
}

### Steenee.java

package pokemons.info;  
import ru.ifmo.se.pokemon.\*;  
import abilities.\*;  
  
public class Steenee extends Bounsweet{  
 public Steenee(String Name, int Lvl){  
 super(Name, Lvl);  
 setStats(52, 40, 48, 40, 48, 62);  
 setType(Type.*GRASS*);  
 setMove(new RazorLeaf(), new Splash(), new PlayNice());  
 }  
}

### Tsareena.java

package pokemons.info;  
import ru.ifmo.se.pokemon.\*;  
import abilities.\*;  
  
public class Tsareena extends Steenee {  
 public Tsareena(String Name, int Lvl){  
 super(Name, Lvl);  
 setStats(72, 120, 98, 50, 98, 72);  
 setType(Type.*GRASS*);  
 setMove(new RazorLeaf(), new Splash(), new PlayNice());  
 }  
}

## Abilities

### Confide.java

package abilities;  
import ru.ifmo.se.pokemon.\*;  
  
public class Confide extends StatusMove {  
 public Confide() {  
 super(Type.*NORMAL*,0,0);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 p.setMod(Stat.*SPECIAL\_ATTACK*, -1);  
 }  
  
 @Override  
 protected String describe(){  
 return "Rонфедерация? нет Confide";  
 }  
  
}

### DarkPulse.java

package abilities;  
import ru.ifmo.se.pokemon.\*;  
  
public class DarkPulse extends SpecialMove {  
 public DarkPulse() {  
 super(Type.*DARK*, 80, 100);  
 }  
  
 private boolean flag;  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if (Math.*random*() <= 0.2) {  
 flag = true;  
 Effect.*flinch*(p);  
 }  
 }  
  
 @Override  
 protected String describe(){  
 if (flag) {return "Темно и страшно пульсирует";}  
 else {return "Темно пульсирует";}  
 }  
}

### DoubleTeam.java

package abilities;  
import ru.ifmo.se.pokemon.\*;  
  
import static ru.ifmo.se.pokemon.Stat.*EVASION*;  
  
public 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 "Повысил ловкость";  
 }  
}

### FlashCannon.java

package abilities;  
import ru.ifmo.se.pokemon.\*;  
  
public class FlashCannon extends SpecialMove {  
 public FlashCannon() {  
 super(Type.*STEEL*, 80,100);  
 }  
  
 private boolean flag;  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if (Math.*random*() <= 0.1){  
 flag = true;  
 p.setMod(Stat.*SPECIAL\_DEFENSE*, -1);  
 }  
 }  
  
 @Override  
 protected String describe(){  
 if (flag) {return "Канонная флешка -1";}  
 else {return "Канонная флешка";}  
 }  
}

### MetalClaw.java

package abilities;  
import ru.ifmo.se.pokemon.\*;  
  
public class MetalClaw extends PhysicalMove {  
 public MetalClaw(){  
 super(Type.*STEEL*, 50,95);  
 }  
  
 private boolean flag;  
 @Override  
 protected void applySelfEffects(Pokemon p){  
 if (Math.*random*() <= 0.1) {  
 flag = true;  
 p.setMod(Stat.*ATTACK*, +1);  
 }  
 }  
  
 @Override  
 protected String describe(){  
 if (flag) {return "Мет коготь +1";}  
 else {return "Мет коготь";}  
 }  
}

### NightSlash.java

package abilities;  
import ru.ifmo.se.pokemon.\*;  
  
public class NightSlash extends PhysicalMove {  
 public NightSlash() {  
 super(Type.*DARK*, 70, 100);  
 }  
  
 private boolean flag;  
 @Override  
 protected double calcCriticalHit(Pokemon att, Pokemon def){  
 if (Math.*random*() <= att.getStat(Stat.*SPEED*) / 512){  
 flag = true;  
 return 2;  
 }  
 else{  
 flag = false;  
 return 1;  
 }  
 }  
  
  
 @Override  
 protected String describe(){  
 if (flag) {return "ННыЫыЫыыААаАаАаа :D";}  
 else {return "не ННыЫыЫыыААаАаАаа :(";}  
 }  
  
}

### PlayNice.java

package abilities;  
import ru.ifmo.se.pokemon.\*;  
  
public class PlayNice extends StatusMove {  
 public PlayNice() {  
 super(Type.*NORMAL*, 0,0);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 p.setMod(Stat.*ATTACK*, -1);  
 }  
  
 @Override  
 protected String describe(){  
 return "Хорошо сыграно, -1";  
 }  
}

### RazorLeaf.java

package abilities;  
import ru.ifmo.se.pokemon.\*;  
  
public class RazorLeaf extends PhysicalMove {  
 public RazorLeaf() {  
 super(Type.*GRASS*, 55,95);  
 }  
  
 private boolean flag;  
 @Override  
 protected double calcCriticalHit(Pokemon att, Pokemon def){  
 if (Math.*random*() <= att.getStat(Stat.*SPEED*) / 512){  
 flag = true;  
 return 2;  
 }  
 else{  
 flag = false;  
 return 1;  
 }  
 }  
  
 @Override  
 protected String describe(){  
 if (flag) {return "ННыЫыЫыыААаАаАаа :D";}  
 else {return "не ННыЫыЫыыААаАаАаа :(";}  
 }  
}

### SludgeBomb.java

package abilities;  
import ru.ifmo.se.pokemon.\*;  
  
public class SludgeBomb extends SpecialMove {  
 public SludgeBomb(){  
 super(Type.*POISON*, 90, 100);  
 }  
  
 private boolean flag;  
 @Override  
 public void applyOppEffects(Pokemon p) {  
 if ((Math.*random*() <= 0.3) & (!p.hasType(Type.*POISON*) | !p.hasType(Type.*STEEL*))){  
 flag = true;  
 Effect.*poison*(p);  
 }  
 }  
  
 @Override  
 protected String describe(){  
 if(flag){return "дамагает и травит";}  
 else{return "дамагает";}  
 }  
}

### Splash.java

package abilities;  
import ru.ifmo.se.pokemon.\*;  
  
public class Splash extends StatusMove {  
 public Splash() {  
 super(Type.*NORMAL*, 0, 0);  
 }  
  
 @Override  
 protected String describe(){  
 return "ничего.., но зачем ты это прожал... тяпа-растяпа";  
 }  
}

### Swagger.java

package abilities;  
import ru.ifmo.se.pokemon.\*;  
  
public class Swagger extends StatusMove {  
 public Swagger(){  
 super(Type.*NORMAL*, 0, 85);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 Effect.*confuse*(p);  
 p.setMod(Stat.*ATTACK*, +2);  
 }  
  
 @Override  
 protected String describe(){  
 return "Свегает";  
 }  
  
}

# Результат выполнения программы

Magearna RoboGirl из команды фиолетовых вступает в бой!

Bounsweet LittleBerry из команды зеленых вступает в бой!

Magearna RoboGirl Канонная флешка.

Bounsweet LittleBerry теряет 6 здоровья.

Bounsweet LittleBerry промахивается

Magearna RoboGirl Мет коготь.

Bounsweet LittleBerry теряет 5 здоровья.

Bounsweet LittleBerry не ННыЫыЫыыААаАаАаа :(.

Magearna RoboGirl теряет 6 здоровья.

Magearna RoboGirl промахивается

Bounsweet LittleBerry промахивается

Magearna RoboGirl Канонная флешка.

Bounsweet LittleBerry теряет 9 здоровья.

Bounsweet LittleBerry теряет сознание.

Steenee MediumBerry из команды зеленых вступает в бой!

Magearna RoboGirl не ННыЫыЫыыААаАаАаа :(.

Steenee MediumBerry теряет 6 здоровья.

Steenee MediumBerry промахивается

Magearna RoboGirl промахивается

Steenee MediumBerry промахивается

Magearna RoboGirl Мет коготь.

Steenee MediumBerry теряет 8 здоровья.

Steenee MediumBerry теряет сознание.

Tsareena BigBerry из команды зеленых вступает в бой!

Tsareena BigBerry не ННыЫыЫыыААаАаАаа :(.

Magearna RoboGirl теряет 3 здоровья.

Magearna RoboGirl Канонная флешка.

Tsareena BigBerry теряет 6 здоровья.

Tsareena BigBerry промахивается

Magearna RoboGirl Мет коготь.

Tsareena BigBerry теряет 7 здоровья.

Tsareena BigBerry теряет сознание.

В команде зеленых не осталось покемонов.

Команда фиолетовых побеждает в этом бою!

# Заключение

По итогу выполнения Лабораторной работы №2 я узнал, что такое объектно-ориентированное программирование, получил новые навыки работы с ООП в Java, научился работать с классами, конструкторами, полями и модификаторами доступа.