Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2

по дисциплине “Программирование”

Вариант 21539

Выполнил: студент группы P3115

Пеньков Георгий Дмитриевич

Преподаватель:

Усков Иван Владимирович

г. Санкт-Петербург,

2021

# Оглавление

[**Задание**](#_wy1i1371kgwy) **3**

[**Диаграмма классов**](#_e8wupkludw4b) **5**

[**Исходный код программы**](#_jwz1uwjr7xc) **6**

[**Результат работы программы**](#_oeetohiqctuw) **12**

[**Вывод**](#_id469wyamccz) **15**

# 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:

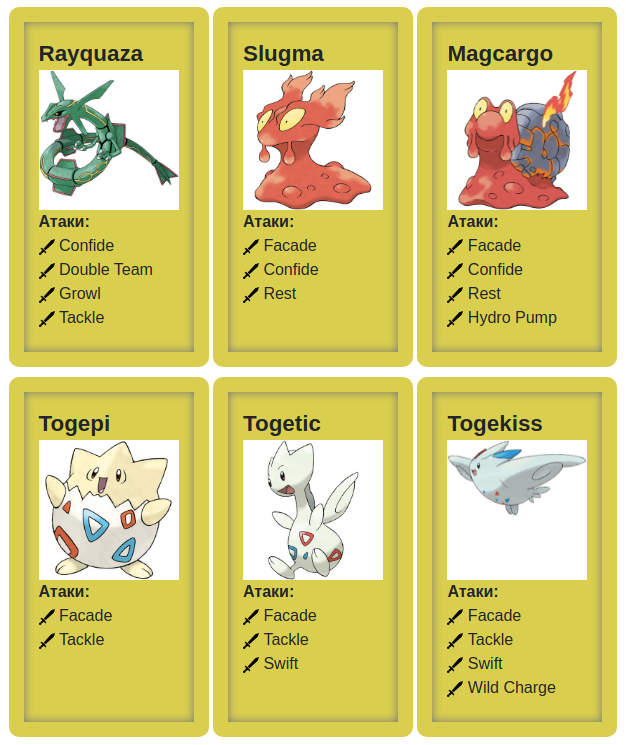
* очки здоровья (HP)
* атака (attack)
* защита (defense)
* специальная атака (special attack)
* специальная защита (special defense)
* скорость (speed)

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.

Мои покемоны:



# Диаграмма классов

Диаграмма классов представлена на рисунке ниже:

# Исходный код программы

[Репозиторий лабораторной работы №2 на github.](https://github.com/penkovgd/java-lab2)

Все покемоны принадлежат пакету pokemons, а атаки пакету moves. Класс labDemo имеет пакет по умолчанию.

#### src/labDemo.java

import pokemons.\*;

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

public class labDemo {

public static void main(String[] args) {

Pokemon[] allies = {

new Slugma("Волк", 3),

new Togetic("Всеволод", 8),

new Togekiss("Владимир", 15)

};

Pokemon[] foes = {

new Togepi("Святослав", 4),

new Magcargo("Радослав", 6),

new Rayquaza("Ростислав", 15)

};

Battle battle = new Battle();

for (Pokemon p : allies) battle.addAlly(p);

for (Pokemon p : foes) battle.addFoe(p);

battle.go();

}

}

#### src/pokemons/Slugma.java

package pokemons;

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

import moves.\*;

public class Slugma extends Pokemon {

public Slugma() {

this("Unnamed", 1);

}

public Slugma(String name, int lvl) {

super(name, lvl);

this.setStats(40.0, 40.0, 70.0, 40.0, 40.0, 20.0);

this.setType(Type.*FIRE*);

this.setMove(new Facade(), new Confide(), new Rest());

}

}

#### src/pokemons/Magcargo.java

package pokemons;

import moves.\*;

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

public class Magcargo extends Slugma {

public Magcargo() {

this("Unnamed", 1);

}

public Magcargo(String name, int lvl) {

super(name, lvl);

this.setStats(60.0, 50.0, 120.0, 90.0, 80.0, 30.0);

this.addType(Type.*ROCK*);

this.addMove(new HydroPump());

}

}

#### 

#### src/pokemons/Rayquaza.java

package pokemons;

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

import moves.\*;

public class Rayquaza extends Pokemon {

public Rayquaza() {

this("Unnamed", 1);

}

public Rayquaza(String name, int lvl) {

super(name, lvl);

this.setStats(105.0, 150.0, 90.0, 150.0, 90.0, 95.0);

this.setType(Type.*DRAGON*, Type.*FLYING*);

this.setMove(new Confide(), new DoubleTeam(), new Growl(), new Tackle());

}

}

#### src/pokemons/Togepi.java

package pokemons;

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

import moves.\*;

public class Togepi extends Pokemon {

public Togepi() {

this("Unnamed", 1);

}

public Togepi(String name, int lvl) {

super(name, lvl);

this.setStats(35.0, 20.0, 65.0, 40.0, 65.0, 20.0);

this.setType(Type.*FAIRY*);

this.setMove(new Facade(), new Tackle());

}

}

#### 

#### src/pokemons/Togetic.java

package pokemons;

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

import moves.\*;

public class Togetic extends Togepi {

public Togetic() {

this("Unnamed", 1);

}

public Togetic(String name, int lvl) {

super(name, lvl);

this.setStats(55.0, 40.0, 85.0, 80.0, 105.0, 40.0);

this.addType(Type.*FLYING*);

this.addMove(new Swift());

}

}

#### src/pokemons/Togekiss.java

package pokemons;

import moves.\*;

public class Togekiss extends Togetic {

public Togekiss() {

this("Unnamed", 1);

}

public Togekiss(String name, int lvl) {

super(name, lvl);

this.setStats(85.0, 50.0, 95.0, 120.0, 115.0, 80.0);

this.addMove(new WildCharge());

}

}

#### 

#### src/moves/Confide.java

package moves;

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

public class Confide extends StatusMove {

*/\* Confide lowers the target's Special Attack by one stage.*

*\* Stats can be lowered to a minimum of -6 stages each.*

*\*/*

public Confide() {

super();

type = Type.*NORMAL*;

}

protected void applyOppEffects(Pokemon p) {

p.setMod(Stat.*SPECIAL\_ATTACK*, -1);

}

protected String describe() {

return "использует Confide";

}

}

#### src/moves/DoubleTeam.java

package moves;

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

public class DoubleTeam extends StatusMove {

*/\* Double Team raises the user's Evasiveness by one stage,*

*\* thus making the user more difficult to hit.*

*\*/*

public DoubleTeam() {

super();

type = Type.*NORMAL*;

}

protected void applySelfEffects(Pokemon p) {

p.setMod(Stat.*EVASION*, 1);

}

protected String describe() {

return "использует Double Team";

}

}

#### src/moves/Facade.java

package moves;

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

public class Facade extends PhysicalMove {

*/\* Facade deals damage, and hits with double power (140)*

*\* if the user is burned, poisoned or paralyzed.*

*\* In the case of a burn, the usual attack-halving still*

*\* occurs so Facade hits with an effective power of 70.*

*\*/*

public Facade() {

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

}

protected double calcBaseDamage(Pokemon att, Pokemon def) {

Status cond = def.getCondition();

if (cond.equals(Status.*BURN*) ||

cond.equals(Status.*POISON*) ||

cond.equals(Status.*PARALYZE*)) {

power \*= 2;

}

return super.calcBaseDamage(att, def);

}

protected String describe() {

return "использует Facade";

}

}

#### 

#### src/moves/Growl.java

package moves;

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

public class Growl extends StatusMove {

*/\* Growl lowers the target's Attack by one stage.*

*\* Stats can be lowered to a minimum of -6 stages each.*

*\*/*

public Growl() {

super();

accuracy = 1.0;

}

protected void applyOppEffects(Pokemon p) {

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

}

protected String describe() {

return "использует Growl";

}

}

#### src/moves/HydroPump.java

package moves;

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

public class HydroPump extends SpecialMove {

*// Hydro Pump deals damage with no additional effect.*

public HydroPump() {

super(Type.*WATER*, 110.0, 0.8);

}

protected String describe() {

return "использует Hydro Pump";

}

}

#### src/moves/Rest.java

package moves;

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

public class Rest extends StatusMove {

*/\* The user goes to sleep for two turns.*

*\* This fully restores the user’s HP and heals any status conditions.*

*\*/*

public Rest() {

super();

type = Type.*PSYCHIC*;

}

protected void applySelfEffects(Pokemon p) {

p.setCondition(new Effect().condition(Status.*SLEEP*).turns(2));

p.restore();

}

protected String describe() {

return "использует Rest";

}

}

#### 

#### src/moves/Swift.java

package moves;

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

public class Swift extends SpecialMove {

*/\* Swift deals damage and ignores changes to the Accuracy and Evasion stats.*

*\* However, it will not hit Pokemon during the invulnerable stage of Bounce,*

*\* Dig, Dive, Fly, Shadow Force or Sky Drop.*

*\*/*

public Swift() {

super(Type.*NORMAL*, 60.0, 1.0);

}

protected boolean checkAccuracy(Pokemon att, Pokemon def) {

return true;

}

protected String describe() {

return "использует Swift";

}

}

#### src/moves/Tackle.java

package moves;

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

public class Tackle extends PhysicalMove {

*/\* Tackle is one of the most common and basic moves a Pokemon learns.*

*\* It deals damage with no additional effects.*

*\*/*

public Tackle() {

super(Type.*NORMAL*, 40.0, 1.0);

}

protected String describe() {

return "использует Tackle";

}

}

#### 

#### src/moves/WildCharge.java

package moves;

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

public class WildCharge extends PhysicalMove {

*/\* Wild Charge deals damage, but the user receives*

*\* 1/4 of the damage it inflicted in recoil.*

*\* In other words, if the attack does 100 HP damage*

*\* to the opponent, the user will lose 25 HP.*

*\*/*

public WildCharge() {

super(Type.*ELECTRIC*, 90.0, 1.0);

}

protected void applySelfDamage(Pokemon att, double damage) {

att.setMod(Stat.*HP*, (int) Math.*round*(damage / 4));

}

protected String describe() {

return "использует Wild Charge";

}

}

#### Дополнение манифеста (doc/manifestAddition)

Main-Class: labDemo

Class-Path: lib/Pokemon.jar

#### Компиляция, сборка jar-архива, запуск (doc/start.sh)

**#!/bin/bash**

javac -cp lib/Pokemon.jar -d classes src/labDemo.java src/\*/\*

jar cfm java-lab2.jar doc/manifestAddition -C classes .

java -jar java-lab2.jar

# 

# Результат работы программы

Slugma Волк из команды красных вступает в бой!

Togepi Святослав из команды синих вступает в бой!

Togepi Святослав использует Tackle.

Критический удар!

Slugma Волк теряет 9 здоровья.

Slugma Волк использует Rest.

Slugma Волк засыпает

Togepi Святослав использует Tackle.

Slugma Волк теряет 4 здоровья.

Slugma Волк использует Confide.

Togepi Святослав уменьшает специальную атаку.

Togepi Святослав использует Facade.

Slugma Волк теряет 5 здоровья.

Slugma Волк использует Confide.

Togepi Святослав уменьшает специальную атаку.

Togepi Святослав использует Facade.

Slugma Волк теряет 6 здоровья.

Slugma Волк использует Confide.

Togepi Святослав уменьшает специальную атаку.

Togepi Святослав использует Tackle.

Slugma Волк теряет 3 здоровья.

Slugma Волк теряет сознание.

Togetic Всеволод из команды красных вступает в бой!

Togetic Всеволод использует Facade.

Togepi Святослав теряет 5 здоровья.

Togepi Святослав использует Tackle.

Togetic Всеволод теряет 3 здоровья.

Togetic Всеволод использует Facade.

Togepi Святослав теряет 9 здоровья.

Togepi Святослав использует Facade.

Togetic Всеволод теряет 3 здоровья.

Togetic Всеволод использует Tackle.

Togepi Святослав теряет 6 здоровья.

Togepi Святослав теряет сознание.

Magcargo Радослав из команды синих вступает в бой!

Togetic Всеволод использует Tackle.

Magcargo Радослав теряет 2 здоровья.

Magcargo Радослав использует Hydro Pump.

Togetic Всеволод теряет 8 здоровья.

Togetic Всеволод использует Tackle.

Magcargo Радослав теряет 2 здоровья.

Magcargo Радослав использует Confide.

Togetic Всеволод уменьшает специальную атаку.

Togetic Всеволод использует Tackle.

Magcargo Радослав теряет 3 здоровья.

Magcargo Радослав использует Hydro Pump.

Togetic Всеволод теряет 8 здоровья.

Togetic Всеволод использует Swift.

Magcargo Радослав теряет 6 здоровья.

Magcargo Радослав использует Confide.

Togetic Всеволод уменьшает специальную атаку.

Togetic Всеволод использует Tackle.

Magcargo Радослав теряет 2 здоровья.

Magcargo Радослав использует Confide.

Togetic Всеволод уменьшает специальную атаку.

Togetic Всеволод использует Facade.

Magcargo Радослав теряет 3 здоровья.

Magcargo Радослав использует Confide.

Togetic Всеволод уменьшает специальную атаку.

Togetic Всеволод использует Swift.

Magcargo Радослав теряет 1 здоровья.

Magcargo Радослав использует Facade.

Togetic Всеволод теряет 4 здоровья.

Togetic Всеволод использует Facade.

Magcargo Радослав теряет 2 здоровья.

Magcargo Радослав использует Facade.

Togetic Всеволод теряет 6 здоровья.

Togetic Всеволод теряет сознание.

Togekiss Владимир из команды красных вступает в бой!

Togekiss Владимир использует Facade.

Magcargo Радослав теряет 6 здоровья.

Magcargo Радослав теряет сознание.

Rayquaza Ростислав из команды синих вступает в бой!

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир использует Facade.

Rayquaza Ростислав теряет 9 здоровья.

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир использует Tackle.

Rayquaza Ростислав теряет 5 здоровья.

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир использует Swift.

Rayquaza Ростислав восстанавливает 8 здоровья.

Rayquaza Ростислав использует Growl.

Togekiss Владимир уменьшает атаку.

Togekiss Владимир использует Facade.

Rayquaza Ростислав теряет 7 здоровья.

Rayquaza Ростислав использует Double Team.

Rayquaza Ростислав увеличивает уклоняемость.

Togekiss Владимир использует Tackle.

Rayquaza Ростислав теряет 6 здоровья.

Rayquaza Ростислав использует Tackle.

Togekiss Владимир теряет 8 здоровья.

Togekiss Владимир использует Tackle.

Rayquaza Ростислав теряет 5 здоровья.

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир использует Tackle.

Rayquaza Ростислав теряет 5 здоровья.

Rayquaza Ростислав использует Growl.

Togekiss Владимир уменьшает атаку.

Togekiss Владимир использует Facade.

Rayquaza Ростислав восстанавливает 1 здоровья.

Rayquaza Ростислав использует Growl.

Togekiss Владимир уменьшает атаку.

Togekiss Владимир использует Facade.

Rayquaza Ростислав восстанавливает 1 здоровья.

Rayquaza Ростислав использует Growl.

Togekiss Владимир уменьшает атаку.

Togekiss Владимир использует Swift.

Rayquaza Ростислав восстанавливает 1 здоровья.

Rayquaza Ростислав использует Growl.

Togekiss Владимир уменьшает атаку.

Togekiss Владимир использует Facade.

Rayquaza Ростислав теряет 3 здоровья.

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир промахивается

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир промахивается

Rayquaza Ростислав использует Double Team.

Rayquaza Ростислав увеличивает уклоняемость.

Togekiss Владимир промахивается

Rayquaza Ростислав использует Growl.

Togekiss Владимир уменьшает атаку.

Togekiss Владимир промахивается

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир использует Swift.

Rayquaza Ростислав теряет 2 здоровья.

Rayquaza Ростислав использует Tackle.

Togekiss Владимир теряет 7 здоровья.

Togekiss Владимир использует Facade.

Rayquaza Ростислав теряет 2 здоровья.

Rayquaza Ростислав использует Tackle.

Togekiss Владимир теряет 10 здоровья.

Togekiss Владимир промахивается

Rayquaza Ростислав использует Tackle.

Togekiss Владимир теряет 5 здоровья.

Togekiss Владимир использует Facade.

Rayquaza Ростислав теряет 4 здоровья.

Rayquaza Ростислав использует Double Team.

Rayquaza Ростислав увеличивает уклоняемость.

Togekiss Владимир использует Facade.

Rayquaza Ростислав теряет 2 здоровья.

Rayquaza Ростислав использует Tackle.

Togekiss Владимир теряет 9 здоровья.

Togekiss Владимир использует Wild Charge.

Rayquaza Ростислав теряет 3 здоровья.

Togekiss Владимир теряет 1 здоровья.

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир использует Swift.

Критический удар!

Rayquaza Ростислав теряет 2 здоровья.

Rayquaza Ростислав использует Growl.

Togekiss Владимир уменьшает атаку.

Togekiss Владимир использует Facade.

Rayquaza Ростислав теряет 2 здоровья.

Rayquaza Ростислав использует Double Team.

Rayquaza Ростислав увеличивает уклоняемость.

Togekiss Владимир промахивается

Rayquaza Ростислав использует Double Team.

Rayquaza Ростислав увеличивает уклоняемость.

Togekiss Владимир промахивается

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир использует Swift.

Rayquaza Ростислав теряет 1 здоровья.

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир использует Facade.

Rayquaza Ростислав теряет 3 здоровья.

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир промахивается

Rayquaza Ростислав использует Double Team.

Rayquaza Ростислав увеличивает уклоняемость.

Togekiss Владимир использует Swift.

Rayquaza Ростислав теряет 1 здоровья.

Rayquaza Ростислав использует Growl.

Togekiss Владимир уменьшает атаку.

Togekiss Владимир использует Wild Charge.

Rayquaza Ростислав теряет 4 здоровья.

Togekiss Владимир теряет 1 здоровья.

Rayquaza Ростислав использует Double Team.

Rayquaza Ростислав увеличивает уклоняемость.

Togekiss Владимир промахивается

Rayquaza Ростислав использует Double Team.

Rayquaza Ростислав увеличивает уклоняемость.

Togekiss Владимир промахивается

Rayquaza Ростислав использует Confide.

Togekiss Владимир уменьшает специальную атаку.

Togekiss Владимир промахивается

Rayquaza Ростислав использует Tackle.

Togekiss Владимир теряет 5 здоровья.

Togekiss Владимир использует Tackle.

Rayquaza Ростислав теряет 4 здоровья.

Rayquaza Ростислав теряет сознание.

В команде синих не осталось покемонов.

Команда красных побеждает в этом бою!

# Вывод

Благодаря лабораторной работе №2 я познакомился с основными принципами ООП и их реализациями в языке java, в том числе: познакомился с механизмом наследования, системе доступа к членам класса и самого класса, переопределением методов, работой с пакетами. Также, я научился подключать внешние jar-файлы к моему проекту и разделять исходный код от скомпилированных классов, в общем, лучше структурировать свой проект. Также я научился создавать uml диаграммы классов с помощью инструментов IntelliJ IDEA.