Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Лабораторная работа № 2**

По дисциплине «Программирование»

Вариант 3117250

Выполнила:

Студентка группы P3117

Русакова Е.Д.

Преподаватель:

Письмак А.Е.

Санкт-Петербург

2022

Содержание

# 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.

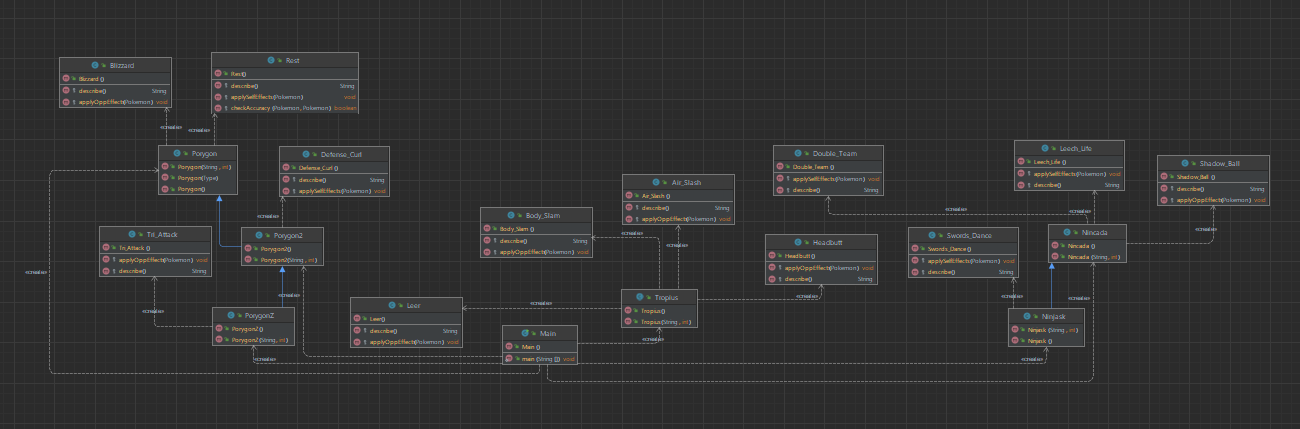
Используя класс симуляции боя Battle, создать 2 команды покемонов (каждый покемон должен иметь имя) и запустить бой.

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>



# Диаграмма классов реализованной объектной модели:



# Исходный код программы:

## Сражение:

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

**import** **pokemon.\***;

**public** **class** **Main** {

**public** **static** **void** **main**(String[] args) {

Battle b = **new** Battle();

Pokemon p1 = **new** Nincada("Нинкада",**10**);

Pokemon p2 = **new** Ninjask("Нинджаcк",**10**);

Pokemon p3 = **new** Porygon("Поригон",**10**);

Pokemon p4 = **new** Porygon2("Поригон-2", **10**);

Pokemon p5 = **new** PorygonZ("Поригон-z",**10**);

Pokemon p6 = **new** Tropius("Тропиус", **10**);

b.addAlly(p1);

b.addAlly(p3);

b.addAlly(p5);

b.addFoe(p2);

b.addFoe(p4);

b.addFoe(p6);

b.go();

}

}

## Покемоны:

### Tropius

**package** pokemon;

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

**import** **moves.\***;

**public** **class** **Tropius** **extends** Pokemon{

**private** **double** hp = **99**;

**private** **double** att = **68**;

**private** **double** def = **83**;

**private** **double** spAtt = **72**;

**private** **double** spDef = **87**;

**private** **double** speed = **51**;

**public** **Tropius**() {

**super**();

setType(Type.GRASS, Type.FLYING);

setStats(hp, att, def, spAtt, spDef, speed);

setMove(**new** Body\_Slam(), **new** Headbutt(), **new** Air\_Slash(), **new** Leer());

}

**public** **Tropius**(String name, **int** level) {

**super**(name, level);

setType(Type.GRASS, Type.FLYING);

setStats(hp, att, def, spAtt, spDef, speed);

setMove(**new** Body\_Slam(), **new** Headbutt(), **new** Air\_Slash(), **new** Leer());

}

}

### Nincada

**package** pokemon;

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

**import** **moves.\***;

**public** **class** **Nincada** **extends** Pokemon{

**private** **double** hp = **31**;

**private** **double** att = **45**;

**private** **double** def = **90**;

**private** **double** spAtt = **30**;

**private** **double** spDef = **30**;

**private** **double** speed = **40**;

**public** **Nincada**() {

**super**();

setType(Type.BUG, Type.GROUND);

setStats(hp, att, def, spAtt, spDef, speed);

setMove(**new** Leech\_Life(), **new** Shadow\_Ball(), **new** Double\_Team());

}

**public** **Nincada**(java.lang.String name, **int** level) {

**super**(name, level);

setStats(hp, att, def, spAtt, spDef, speed);

setMove(**new** Leech\_Life(), **new** Shadow\_Ball(), **new** Double\_Team());

}

}

### Ninjask

**package** pokemon;

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

**import** **moves.\***;

**public** **class** **Ninjask** **extends** Nincada{

**private** **double** hp = **61**;

**private** **double** att = **90**;

**private** **double** def = **45**;

**private** **double** spAtt = **50**;

**private** **double** spDef = **50**;

**private** **double** speed = **160**;

**public** **Ninjask**() {

**super**();

setType(Type.BUG, Type.FLYING);

setStats(hp, att, def, spAtt, spDef, speed);

addMove(**new** Swords\_Dance());

}

**public** **Ninjask**(String name, **int** level) {

**super**(name, level);

setType(Type.BUG, Type.FLYING);

setStats(hp, att, def, spAtt, spDef, speed);

addMove(**new** Swords\_Dance());

}

}

### Porygon

**package** pokemon;

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

**import** **moves.\***;

**public** **class** **Porygon** **extends** Pokemon {

**private** **double** hp = **65**;

**private** **double** att = **60**;

**private** **double** def = **70**;

**private** **double** spAtt = **85**;

**private** **double** spDef = **75**;

**private** **double** speed = **40**;

**public** **Porygon**() {

**super**();

setType(Type.NORMAL);

setStats(hp, att, def, spAtt, spDef, speed);

setMove(**new** Rest(), **new** Blizzard());

}

**public** **Porygon**(java.lang.String name, **int** level) {

**super**(name, level);

setType(Type.NORMAL);

setStats(hp, att, def, spAtt, spDef, speed);

setMove(**new** Rest(), **new** Blizzard());

}

**public** **Porygon**(Type type){

**this**();

}

}

### Porygon2

**package** pokemon;

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

**import** **moves.\***;

**public** **class** **Porygon2** **extends** Porygon{

**private** **double** hp = **85**;

**private** **double** att = **80**;

**private** **double** def = **90**;

**private** **double** spAtt = **105**;

**private** **double** spDef = **95**;

**private** **double** speed = **60**;

**public** **Porygon2**() {

**super**();

setStats(hp, att, def, spAtt, spDef, speed);

addMove(**new** Defense\_Curl());

}

**public** **Porygon2**(String name, **int** level) {

**super**(name, level);

setStats(hp, att, def, spAtt, spDef, speed);

addMove(**new** Defense\_Curl());

}

}

### Porygon-Z

**package** pokemon;

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

**import** **moves.\***;

**public** **class** **PorygonZ** **extends** Porygon2{

**private** **double** hp = **85**;

**private** **double** att = **80**;

**private** **double** def = **70**;

**private** **double** spAtt = **135**;

**private** **double** spDef = **75**;

**private** **double** speed = **90**;

**public** **PorygonZ**() {

**super**();

setStats(hp, att, def, spAtt, spDef, speed);

addMove(**new** Tri\_Attack());

}

**public** **PorygonZ**(String name, **int** level) {

**super**(name, level);

setStats(hp, att, def, spAtt, spDef, speed);

addMove(**new** Tri\_Attack());

}

}

## Атаки:

### Body Slam

**package** moves;

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

**public** **class** **Body\_Slam** **extends** PhysicalMove{

**public** **Body\_Slam**(){

**super**(Type.NORMAL, **85**, **100**);

}

**@Override**

**protected** **void** **applyOppEffects**(Pokemon p){

Effect e = **new** Effect().chance(**0.3**).turns(**1**).condition(Status.PARALYZE);

p.addEffect(e);

}

**@Override**

**protected** String **describe**(){

**return** "сильно бьет противника, возможно до паралича";

}

}

### Headbutt

**package** moves;

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

**public** **class** **Headbutt** **extends** PhysicalMove{

**public** **Headbutt**(){

**super**(Type.NORMAL, **70**, **100**);

}

**@Override**

**protected** **void** **applyOppEffects** (Pokemon p){

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

Effect.flinch(p);

}

}

**@Override**

**protected** String **describe**(){

**return** "бодается и пугает противника";

}

}

### Air Slash

**package** moves;

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

**public** **class** **Air\_Slash** **extends** SpecialMove{

**public** **Air\_Slash**(){

**super**(Type.FLYING, **75**, **95**);

}

**@Override**

**protected** **void** **applyOppEffects**(Pokemon p){

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

Effect.flinch(p);

}

}

**@Override**

**protected** String **describe**(){

**return** "наносит воздушный удар, чем пугает противника";

}

}

### Leer

**package** moves;

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

**public** **class** **Leer** **extends** StatusMove{

**public** **Leer**(){

**super**();

}

**@Override**

**protected** **void** **applyOppEffects**(Pokemon p){

Effect e = **new** Effect().chance(**1**).stat(Stat.DEFENSE, -**1**);

p.addEffect(e);

}

**@Override**

**protected** String **describe**(){

**return** "хитрит и снижает защиту противника";

}

}

### Leech Life

**package** moves;

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

**public** **class** **Leech\_Life** **extends** PhysicalMove{

**public** **Leech\_Life**(){

**super**(Type.BUG, **80**, **100**);

}

**@Override**

**protected** **void** **applySelfEffects**(Pokemon p){

p.setMod(Stat.HP, Stat.HP.ordinal());

}

**@Override**

**protected** String **describe**(){

**return** "восстанавливает здоровье";

}

}

### Shadow Ball

**package** moves;

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

**public** **class** **Shadow\_Ball** **extends** SpecialMove{

**public** **Shadow\_Ball**(){

**super**(Type.GHOST, **80**, **100**);

}

**@Override**

**protected** **void** **applyOppEffects**(Pokemon p){

Effect e = **new** Effect().turns(**1**).chance(**0.2**).stat(Stat.SPECIAL\_DEFENSE,-**1**);

p.addEffect(e);

}

**@Override**

**protected** String **describe**(){

**return** "использует теневой шар, уменьшает защиту противника";

}

}

### Double Team

**package** moves;

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

**public** **class** **Double\_Team** **extends** StatusMove{

**public** **Double\_Team**(){

**super**();

}

**@Override**

**protected** **void** **applySelfEffects** (Pokemon p){

Effect e = **new** Effect().chance(**1**).turns(**1**).stat(Stat.EVASION, +**1**);

p.addEffect(e);

}

**@Override**

**protected** String **describe**(){

**return** "уклоняется от атаки";

}

}

### Swords Dance

**package** moves;

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

**public** **class** **Swords\_Dance** **extends** StatusMove{

**public** **Swords\_Dance**(){

**super**();

}

**@Override**

**protected** **void** **applySelfEffects**(Pokemon p){

Effect e = **new** Effect().chance(**1**).turns(**1**).stat(Stat.ATTACK, +**2**);

}

**@Override**

**protected** String **describe**(){

**return** "повышает свою атаку";

}

}

### Rest

**package** moves;

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

**public** **class** **Rest** **extends** StatusMove{

**public** **Rest**(){

**super**();

}

**@Override**

**protected** **void** **applySelfEffects**(Pokemon p){

Effect e = **new** Effect().condition(Status.SLEEP).turns(**2**);

p.addEffect(e);

p.setMod(Stat.HP, Stat.HP.ordinal());

}

**@Override**

**protected** **boolean** **checkAccuracy** (Pokemon att,Pokemon def){

**return** **true**;

}

**@Override**

**protected** String **describe**(){

**return** "лечится сном";

}

}

### Blizzard

**package** moves;

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

**public** **class** **Blizzard** **extends** SpecialMove{

**public** **Blizzard**(){

**super**(Type.ICE, **110**, **70**);

}

**@Override**

**protected** **void** **applyOppEffects**(Pokemon p){

Effect e = **new** Effect().chance(**0.1**).condition(Status.FREEZE);

p.addEffect(e);

}

**@Override**

**protected** String **describe**(){

**return** "замораживает цель метелью";

}

}

### Defence Curl

**package** moves;

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

**public** **class** **Defense\_Curl** **extends** StatusMove{

**public** **Defense\_Curl**(){

**super**();

}

**@Override**

**protected** **void** **applySelfEffects**(Pokemon p){

Effect e = **new** Effect().turns(**1**).stat(Stat.DEFENSE, +**1**);

}

**@Override**

**protected** String **describe**(){

**return** "повышает себе защиту";

}

}

### Tri Attack

**package** moves;

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

**public** **class** **Tri\_Attack** **extends** SpecialMove{

**public** **Tri\_Attack**(){

**super**(Type.NORMAL, **80**, **100**);

}

**@Override**

**protected** **void** **applyOppEffects**(Pokemon p){

**double** a = Math.random();

**if**(a < **0.67**) Effect.paralyze(p);

**if**(**0.66** < a && a < **13.34**) Effect.burn(p);

**if**(**13.33** < a && a < **20.01**) Effect.freeze(p);

}

**@Override**

**protected** String **describe**(){

**return** "наносит тройную атаку";

}

}

# Результат работы программы:

**[s367519@helios ~/laba\_2/src]$** java -jar laba\_2.jar

Nincada Нинкада из команды полосатых вступает в бой!

Ninjask Нинджаcк из команды красных вступает в бой!

Ninjask Нинджаcк уклоняется от атаки.

Nincada Нинкада промахивается

Ninjask Нинджаcк уклоняется от атаки.

Nincada Нинкада восстанавливает здоровье.

Ninjask Нинджаcк теряет 4 здоровья.

Ninjask Нинджаcк повышает свою атаку.

Nincada Нинкада уклоняется от атаки.

Ninjask Нинджаcк уклоняется от атаки.

Nincada Нинкада восстанавливает здоровье.

Ninjask Нинджаcк теряет 4 здоровья.

Ninjask Нинджаcк восстанавливает здоровье.

Nincada Нинкада теряет 11 здоровья.

Nincada Нинкада использует теневой шар, уменьшает защиту противника.

Ninjask Нинджаcк теряет 6 здоровья.

Ninjask Нинджаcк уклоняется от атаки.

Nincada Нинкада использует теневой шар, уменьшает защиту противника.

Ninjask Нинджаcк теряет 8 здоровья.

Ninjask Нинджаcк уклоняется от атаки.

Nincada Нинкада использует теневой шар, уменьшает защиту противника.

Ninjask Нинджаcк теряет 7 здоровья.

Ninjask Нинджаcк использует теневой шар, уменьшает защиту противника.

Nincada Нинкада теряет 10 здоровья.

Nincada Нинкада восстанавливает здоровье.

Ninjask Нинджаcк теряет 4 здоровья.

Ninjask Нинджаcк повышает свою атаку.

Nincada Нинкада восстанавливает здоровье.

Ninjask Нинджаcк теряет 4 здоровья.

Ninjask Нинджаcк теряет сознание.

Porygon2 Поригон-2 из команды красных вступает в бой!

Porygon2 Поригон-2 повышает себе защиту.

Nincada Нинкада восстанавливает здоровье.

Porygon2 Поригон-2 теряет 4 здоровья.

Porygon2 Поригон-2 повышает себе защиту.

Nincada Нинкада использует теневой шар, уменьшает защиту противника.

Porygon2 Поригон-2 теряет 1 здоровья.

Porygon2 Поригон-2 не замечает воздействие типа GHOST

Porygon2 Поригон-2 повышает себе защиту.

Nincada Нинкада уклоняется от атаки.

Porygon2 Поригон-2 лечится сном.

Porygon2 Поригон-2 засыпает

Nincada Нинкада восстанавливает здоровье.

Porygon2 Поригон-2 теряет 4 здоровья.

Porygon2 Поригон-2 лечится сном.

Nincada Нинкада уклоняется от атаки.

Porygon2 Поригон-2 повышает себе защиту.

Nincada Нинкада уклоняется от атаки.

Porygon2 Поригон-2 замораживает цель метелью.

Nincada Нинкада теряет 22 здоровья.

Nincada Нинкада замерзает

Nincada Нинкада теряет сознание.

Porygon Поригон из команды полосатых вступает в бой!

Porygon2 Поригон-2 лечится сном.

Porygon2 Поригон-2 засыпает

Porygon Поригон замораживает цель метелью.

Porygon2 Поригон-2 теряет 11 здоровья.

Porygon2 Поригон-2 повышает себе защиту.

Porygon Поригон замораживает цель метелью.

Porygon2 Поригон-2 теряет 8 здоровья.

Porygon2 Поригон-2 повышает себе защиту.

Porygon Поригон замораживает цель метелью.

Porygon2 Поригон-2 теряет 10 здоровья.

Porygon2 Поригон-2 повышает себе защиту.

Porygon Поригон замораживает цель метелью.

Porygon2 Поригон-2 теряет 8 здоровья.

Porygon2 Поригон-2 теряет сознание.

Tropius Тропиус из команды красных вступает в бой!

Tropius Тропиус наносит воздушный удар, чем пугает противника.

Porygon Поригон теряет 9 здоровья.

Porygon Поригон замораживает цель метелью.

Tropius Тропиус теряет 28 здоровья.

Tropius Тропиус сильно бьет противника, возможно до паралича.

Porygon Поригон теряет 6 здоровья.

Porygon Поригон замораживает цель метелью.

Tropius Тропиус теряет 46 здоровья.

Tropius Тропиус теряет сознание.

В команде красных не осталось покемонов.

Команда полосатых побеждает в этом бою!

# Вывод:

При выполнении лабораторной работы я познакомилась с основными принципами объекто-ориентированного программирования, попробовала подключать сторонние jar-файлы к своей программе, познакомилась с переопределениями методов, наследованием классов и системой пакетов.