Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И  
ОПТИКИ  
Факультет систем управления и робототехники

Отчёт по лабораторной работе №2

“Программирование”

Вариант №314214

Преподаватель:

Сорокин Роман Борисович

Выполнила:

Парфенова Ольга Сергеевна

Гр. R3142

Санкт-Петербург, 2020

1. Текст задания

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:

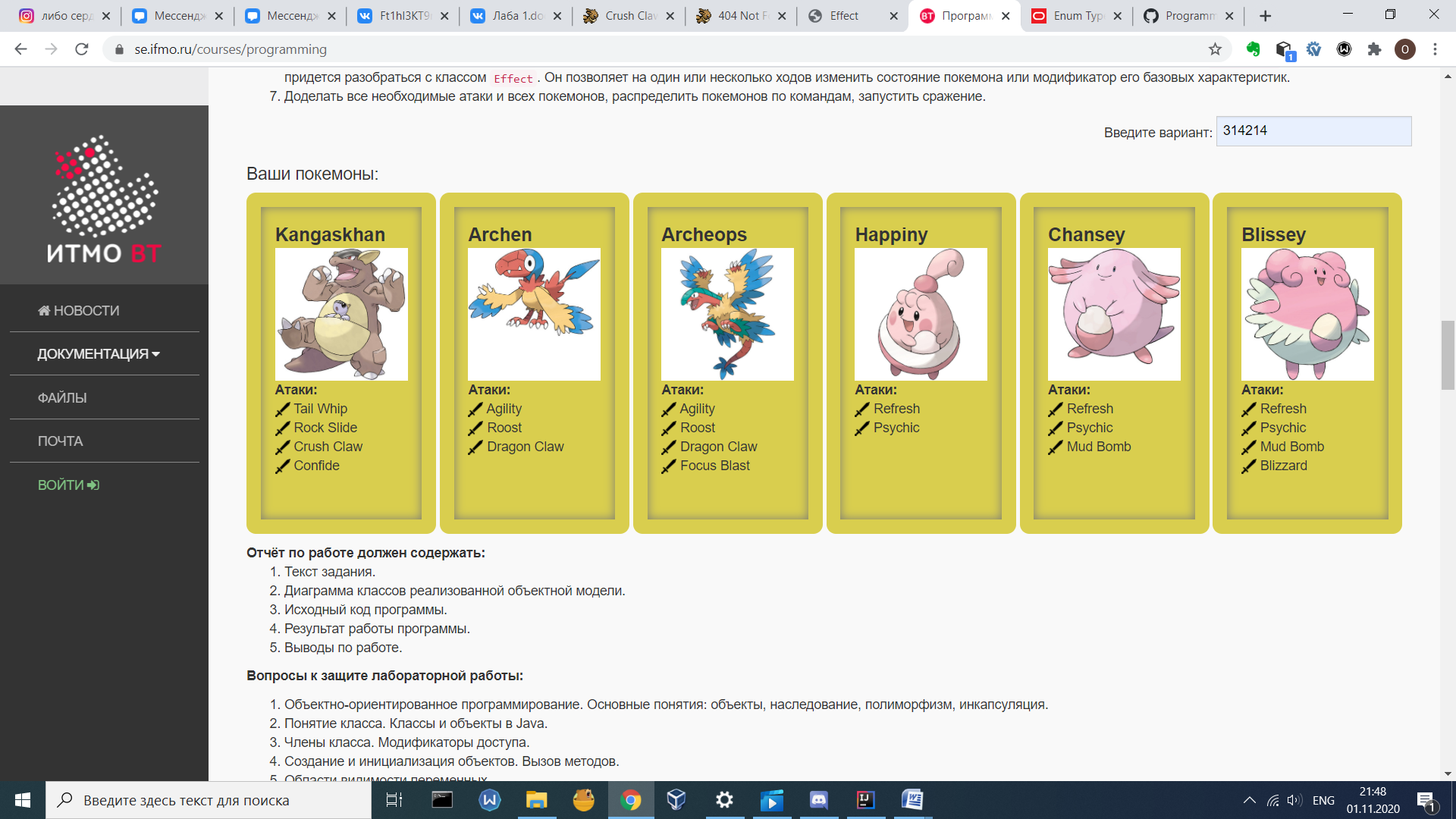
* очки здоровья (HP)
* атака (attack)
* защита (defense)
* специальная атака (special attack)
* специальная защита (special defense)
* скорость (speed)

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.

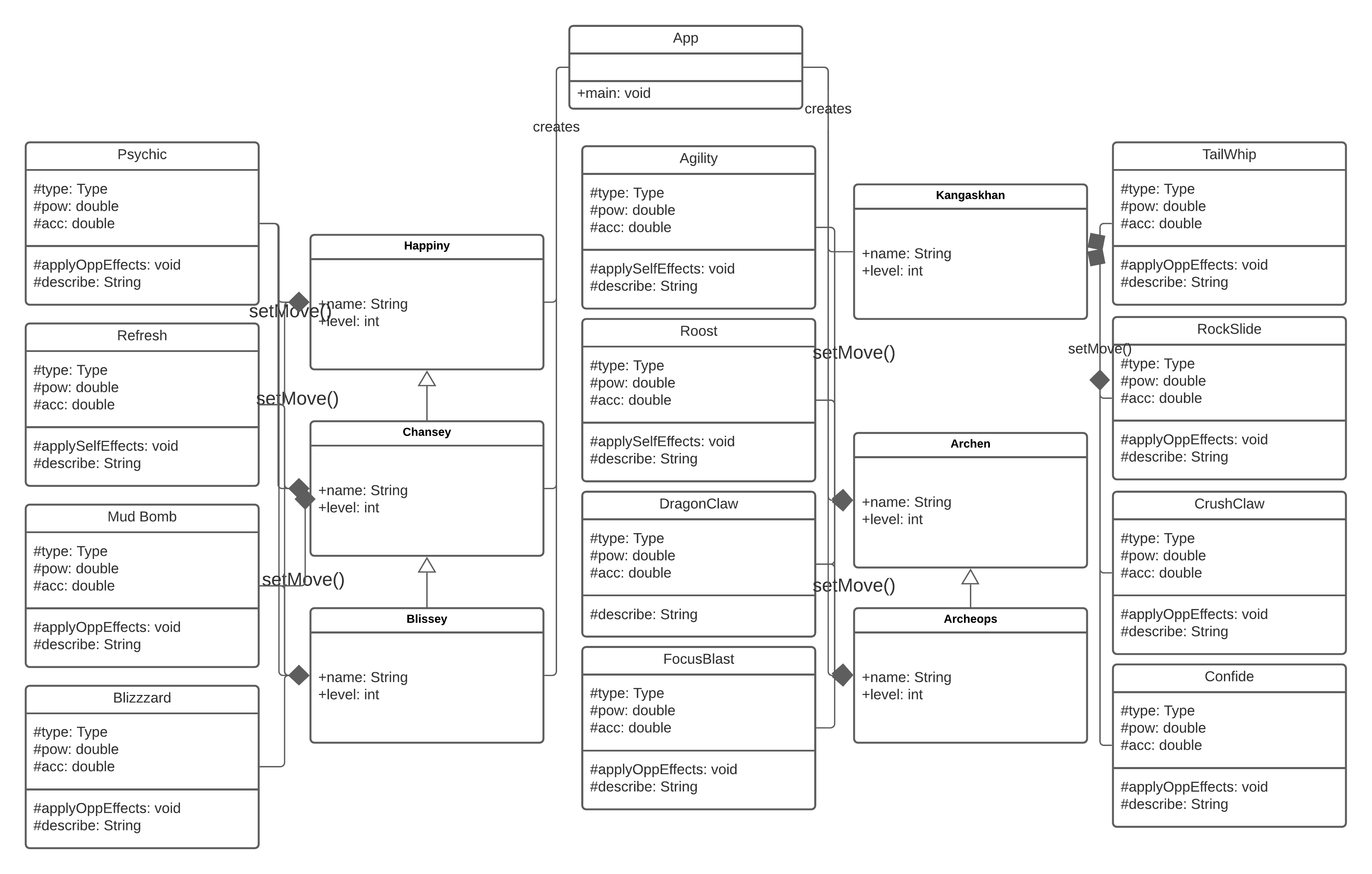
Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атаки. Назначить каждому виду покемонов атаки в соответствии с вариантом. Уровень покемона выбирается минимально необходимым для всех реализованных атак.

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

Мои покемоны:



2. Диаграмма классов



3. Исходный код программы

**App.java**

package Lab2\_gradle;

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

public class App {

public static void main(String[] args) {

Battle b = new Battle();

b.addAlly(new Happiny("Happiny", 1));

b.addAlly(new Archeops("Archeops", 1));

b.addAlly(new Kangaskhan("Kangaskhan", 1));

b.addFoe(new Chansey("Chansey", 1));

b.addFoe(new Archen("Archen", 1));

b.addFoe(new Blissey("Blissey", 1));

b.go();

}

}

**Pokemons.java**

package Lab2\_gradle;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

class Happiny extends Pokemon {

public Happiny(String name, int level) {

super(name, level);

setStats(100, 5, 5, 15, 65, 30);

setType(Type.NORMAL);

setMove(new Refresh(), new Psychic());

}

}

class Chansey extends Happiny {

public Chansey(String name, int level) {

super(name, level);

setStats(250, 5, 5, 35, 105, 50);

setType(Type.NORMAL);

setMove(new Refresh(), new Psychic(), new MudBomb());

}

}

class Blissey extends Chansey {

public Blissey(String name, int level) {

super(name, level);

setStats(255, 10, 10, 75, 135, 55);

setType(Type.NORMAL);

setMove(new Refresh(), new Psychic(), new MudBomb(), new Blizzard());

}

}

class Archen extends Pokemon {

public Archen(String name, int level) {

super(name, level);

setStats(55, 112, 45, 74, 45, 70);

setType(Type.ROCK, Type.FLYING);

setMove(new Agility(), new Roost(), new DragonClaw());

}

}

class Archeops extends Archen {

public Archeops(String name, int level) {

super(name, level);

setStats(75, 140, 65, 112, 65, 110);

setType(Type.FLYING, Type.ROCK);

setMove(new Agility(), new Roost(), new DragonClaw(), new FocusBlast());

}

}

class Kangaskhan extends Pokemon {

public Kangaskhan(String name, int level) {

super(name, level);

setStats(105, 95, 80, 40, 80, 90);

setType(Type.NORMAL);

setMove(new TailWhip(), new RockSlide(), new CrushClaw(), new Confide(), new Confide());

}

}

**BlisseyAttacks.java**

package Lab2\_gradle;

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

class Psychic extends SpecialMove {

protected Psychic() {

super(Type.PSYCHIC, 90, 100);

}

protected void applyOppEffects(Pokemon pokemon) {

Effect e = new Effect().chance(0.1).stat(Stat.SPECIAL\_DEFENSE, -1);

pokemon.addEffect(e);

}

protected String describe() {

return ("uses Psychic");

}

}

class Refresh extends StatusMove {

protected Refresh() {

super(Type.NORMAL, 0, 100);

}

protected void applySelfEffects(Pokemon pokemon) {

Status c = pokemon.getCondition();

Effect e = new Effect().condition(Status.NORMAL);

if (c.equals(Status.PARALYZE) || c.equals(Status.BURN) || c.equals(Status.POISON)) pokemon.addEffect(e);

}

protected String describe() {

return ("uses Refresh");

}

}

class MudBomb extends SpecialMove {

protected MudBomb() {

super(Type.GROUND, 65, 85);

}

protected void applyOppEffects(Pokemon pokemon) {

Effect e = new Effect().chance(0.3).stat(Stat.ACCURACY, -1);

pokemon.addEffect(e);

}

protected String describe() {

return ("uses Mud Bomb");

}

}

class Blizzard extends SpecialMove {

protected Blizzard() {

super(Type.ICE, 110, 70);

}

protected void applyOppEffects(Pokemon pokemon) {

Effect e = new Effect().chance(0.1);

Effect.freeze(pokemon);

}

protected String describe() {

return ("uses Blizzard");

}

}

**ArcheopsAttacks.java**

package Lab2\_gradle;

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

class Agility extends StatusMove {

protected Agility() {

super(Type.PSYCHIC, 0, 100);

}

protected void applySelfEffects(Pokemon pokemon) {

pokemon.setMod(Stat.SPEED, 2);

}

protected String describe() {

return ("uses Agility");

}

}

class Roost extends StatusMove {

protected Roost() {

super(Type.FLYING, 0, 100);

}

protected void applySelfEffects(Pokemon pokemon) {

double max\_hp = pokemon.getStat(Stat.HP);

pokemon.setMod(Stat.HP, (int) -max\_hp / 2);

}

protected String describe() {

return ("uses Roost");

}

}

class DragonClaw extends PhysicalMove {

protected DragonClaw() {

super(Type.DRAGON, 80, 100);

}

protected String describe() {

return ("uses Dragon Claw");

}

}

class FocusBlast extends SpecialMove {

protected FocusBlast() {

super(Type.FIGHTING, 120, 70);

}

protected void applyOppEffects(Pokemon pokemon) {

Effect e = new Effect().chance(0.1).stat(Stat.SPECIAL\_DEFENSE, -1);

pokemon.addEffect(e);

}

protected String describe() {

return ("uses Focus Blast");

}

}

**KangaskhanAttacks.java**

package Lab2\_gradle;

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

class TailWhip extends StatusMove {

protected TailWhip() {

super(Type.NORMAL, 0, 100);

}

protected void applyOppEffects(Pokemon pokemon) {

pokemon.setMod(Stat.DEFENSE, -1);

}

protected String describe() {

return ("uses Tail Whip");

}

}

class RockSlide extends PhysicalMove {

protected RockSlide() {

super(Type.ROCK, 75, 90);

}

protected void applyOppEffects(Pokemon pokemon) {

Effect e = new Effect().chance(0.3);

e.flinch(pokemon);

}

protected String describe() {

return ("uses Rock Slide");

}

}

class CrushClaw extends PhysicalMove {

protected CrushClaw() {

super(Type.NORMAL, 75, 95);

}

protected void applyOppEffects(Pokemon pokemon) {

Effect e = new Effect().chance(0.5).stat(Stat.DEFENSE, -1);

pokemon.addEffect(e);

}

protected String describe() {

return ("uses Crush Claw");

}

}

class Confide extends StatusMove {

protected Confide() {

super(Type.NORMAL, 0, 100);

}

protected void applyOppEffects(Pokemon pokemon) {

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

}

protected String describe() {

return ("uses Confide");

}

}

4. Результат работы программы

Happiny Happiny из команды черных вступает в бой!

Chansey Chansey из команды фиолетовых вступает в бой!

Chansey Chansey uses Psychic.

Happiny Happiny теряет 4 здоровья.

Happiny Happiny uses Refresh.

Chansey Chansey uses Refresh.

Happiny Happiny uses Refresh.

Chansey Chansey uses Psychic.

Happiny Happiny теряет 5 здоровья.

Happiny Happiny uses Refresh.

Chansey Chansey uses Mud Bomb.

Happiny Happiny теряет 6 здоровья.

Happiny Happiny теряет сознание.

Archeops Archeops из команды черных вступает в бой!

Archeops Archeops uses Agility.

Archeops Archeops увеличивает скорость.

Chansey Chansey uses Psychic.

Archeops Archeops теряет 4 здоровья.

Archeops Archeops uses Roost.

Archeops Archeops восстанавливает 6 здоровья.

Chansey Chansey uses Refresh.

Archeops Archeops uses Dragon Claw.

Chansey Chansey теряет 4 здоровья.

Chansey Chansey uses Refresh.

Archeops Archeops uses Roost.

Archeops Archeops восстанавливает 6 здоровья.

Chansey Chansey uses Psychic.

Archeops Archeops теряет 5 здоровья.

Archeops Archeops uses Focus Blast.

Chansey Chansey теряет 14 здоровья.

Chansey Chansey теряет сознание.

Archen Archen из команды фиолетовых вступает в бой!

Archeops Archeops uses Roost.

Archeops Archeops восстанавливает 6 здоровья.

Archen Archen uses Roost.

Archen Archen восстанавливает 6 здоровья.

Archeops Archeops uses Dragon Claw.

Archen Archen теряет 5 здоровья.

Archen Archen uses Dragon Claw.

Archeops Archeops теряет 5 здоровья.

Archeops Archeops uses Focus Blast.

Archen Archen теряет 7 здоровья.

Archen Archen uses Agility.

Archen Archen увеличивает скорость.

Archeops Archeops uses Focus Blast.

Archen Archen теряет 7 здоровья.

Archen Archen теряет сознание.

Blissey Blissey из команды фиолетовых вступает в бой!

Archeops Archeops uses Roost.

Archeops Archeops восстанавливает 6 здоровья.

Blissey Blissey uses Blizzard.

Archeops Archeops теряет 12 здоровья.

Archeops Archeops замерзает

Archeops Archeops оттаивает

Archeops Archeops uses Focus Blast.

Blissey Blissey теряет 9 здоровья.

Blissey Blissey uses Mud Bomb.

Archeops Archeops теряет 1 здоровья.

Archeops Archeops не замечает воздействие типа GROUND

Archeops Archeops uses Agility.

Archeops Archeops увеличивает скорость.

Blissey Blissey uses Psychic.

Archeops Archeops теряет 5 здоровья.

Archeops Archeops uses Focus Blast.

Blissey Blissey теряет 10 здоровья.

Blissey Blissey теряет сознание.

В команде фиолетовых не осталось покемонов.

Команда черных побеждает в этом бою!

5. Выводы

В ходе работы над данной программой я узнала основы объектно-ориентированного программирования, научилась работать с классами, их методами и модификаторами доступа, модификаторами final и static, импортировать внешние jar-файлы и использовать систему сборки Gradle.