МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Национальный исследовательский университет ИТМО

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА № 2**

по дисциплине

«ПРОГРАММИРОВАНИЕ»

Вариант № 88576

**Выполнил:**

Студент группы P3116

Ткачев Илья

Андреевич

**Преподаватель:**

Письмак Алексей

Евгеньевич

Санкт-Петербург, 2022

Содержание

Задание ……………………………………………………………………. 3

Диаграмма ………………………………………………………………… 5

Код программы …………………………………………………………… 6

Результат работы программы …………………………………………... 14

Вывод ……………………………………………………….…...……….. 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.

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>

#### **Комментарии**

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

Что надо сделать (краткое описание)

1. 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.
2. 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.
3. 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);

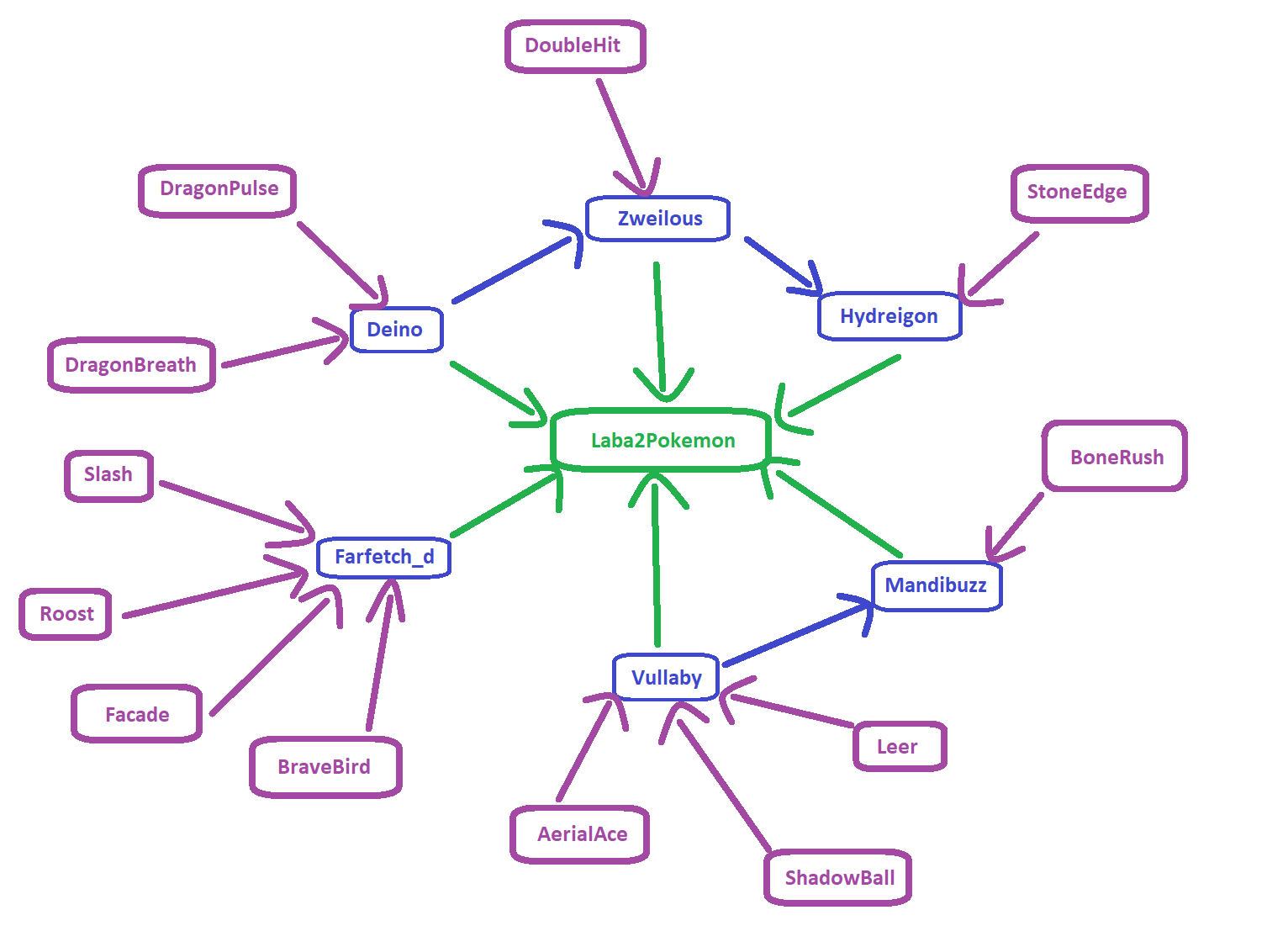
b.go();

1. 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.
2. 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.
3. 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.
4. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

Покемоны для вапианта 88576:



Диаграмма



Код программы

*Laba2Pokemon.java*

import Pokemons.Farfetch\_d;

import Pokemons.Vullaby;

import Pokemons.Mandibuzz;

import Pokemons.Deino;

import Pokemons.Zweilous;

import Pokemons.Hydreigon;

import Lib.Mes;

import ru.ifmo.se.pokemon.Battle;

public class Laba2Pokemon {

public static void main (String[] args) {

Battle battle = new Battle();

Farfetch\_d Tolya = new Farfetch\_d(new String(Mes.Tolya), 1);

Vullaby Voryga = new Vullaby("Voryga", 1);

Mandibuzz Pikachu = new Mandibuzz("Pikachu", 1);

Deino Vafla = new Deino(new String(Mes.Vafla), 1);

Zweilous Mortis = new Zweilous(new String(Mes.Mortis), 1);

Hydreigon Fermer = new Hydreigon(new String(Mes.Fermer), 1);

battle.addAlly(Fermer);

battle.addAlly(Voryga);

battle.addAlly(Pikachu);

battle.addFoe(Vafla);

battle.addFoe(Mortis);

battle.addFoe(Tolya);

battle.go();

}

}

*Deino.java*

*package Pokemons;*

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

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

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

*import Moves.DragonPulse;*

*import Moves.DragonBreath;*

*public class Deino extends Pokemon {*

*public Deino(String var1, int var2) {*

*super(var1, var2);*

*this.setStats(52, 65, 50, 45, 50, 38);*

*this.setType(new Type[]{Type.DARK, Type.DRAGON});*

*this.setMove(new Move[]{new DragonPulse(), new DragonBreath()});*

*}*

*}*

*Farfetch\_d.java*

*package Pokemons;*

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

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

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

*import Moves.Roost;*

*import Moves.Slash;*

*import Moves.BraveBird;*

*import Moves.Facade;*

*public class Farfetch\_d extends Pokemon {*

*public Farfetch\_d(String var1, int var2) {*

*super(var1, var2);*

*this.setStats(52, 90, 55, 58, 62, 60);*

*this.setType(new Type[]{Type.NORMAL, Type.FLYING});*

*this.setMove(new Move[]{new Slash(), new Roost(), new BraveBird(), new Facade()});*

*}*

*}*

*Hydreigon.java*

*package Pokemons;*

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

*import Moves.StoneEdge;*

*public class Hydreigon extends Zweilous {*

*public Hydreigon(String var1, int var2) {*

*super(var1, var2);*

*this.setStats(92, 105, 90, 125, 90, 98);*

*this.setType(new Type[]{Type.DARK, Type.DRAGON});*

*this.addMove(new StoneEdge());*

*}*

*}*

*Mandibuzz.java*

*package Pokemons;*

*import Pokemons.Vullaby;*

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

*import Moves.BoneRush;*

*public class Mandibuzz extends Vullaby {*

*public Mandibuzz(String var1, int var2) {*

*super(var1, var2);*

*this.setStats(110, 65, 105, 55, 95, 80);*

*this.setType(new Type[]{Type.DARK, Type.FLYING});*

*this.addMove(new BoneRush());*

*}*

*}*

*Vullaby.java*

*package Pokemons;*

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

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

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

*import Moves.AerialAce;*

*import Moves.Leer;*

*import Moves.ShadowBall;*

*public class Vullaby extends Pokemon {*

*public Vullaby(String var1, int var2) {*

*super(var1, var2);*

*this.setStats(70, 55, 75, 45, 65, 60);*

*this.setType(new Type[]{Type.DARK, Type.FLYING});*

*this.setMove(new Move[]{new AerialAce(), new Leer(), new ShadowBall()});*

*}*

*}*

*Zweilous.java*

*package Pokemons;*

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

*import Moves.DoubleHit;*

*public class Zweilous extends Deino {*

*public Zweilous(String var1, int var2) {*

*super(var1, var2);*

*this.setStats(72, 85, 70, 65, 70, 58);*

*this.setType(new Type[]{Type.DARK, Type.DRAGON});*

*this.addMove(new DoubleHit());*

*}*

*}*

*AerialAce.java*

*package Moves;*

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

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

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

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

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

*import Lib.Mes;*

*public class AerialAce extends PhysicalMove {*

*public AerialAce() { super(Type.FLYING, 60, 0); }*

*protected void applySelfEffects(Pokemon var1) {*

*Effect var2 = (new Effect()).turns(0).stat(Stat.ACCURACY, 100);*

*var1.addEffect(var2);*

*}*

*protected String describe() { return (new String(Mes.use) + " Aerial Ace"); }*

*}*

*BoneRush.java*

*package Moves;*

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

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

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

*import Lib.Mes;*

*public class BoneRush extends PhysicalMove {*

*public BoneRush() { super(Type.GROUND, 25, 90); }*

*protected void applyOppDamage(Pokemon var1, double var2) {*

*int var3 = (int)(Math.random() \* 4 + 2);*

*if (var3 == 2) {*

*applyOppDamage(var1, var2);*

*applyOppDamage(var1, var2);*

*}*

*else if (var3 == 3) {*

*applyOppDamage(var1, var2);*

*applyOppDamage(var1, var2);*

*applyOppDamage(var1, var2);*

*}*

*else if (var3 == 4) {*

*applyOppDamage(var1, var2);*

*applyOppDamage(var1, var2);*

*applyOppDamage(var1, var2);*

*applyOppDamage(var1, var2);*

*}*

*else {*

*applyOppDamage(var1, var2);*

*applyOppDamage(var1, var2);*

*applyOppDamage(var1, var2);*

*applyOppDamage(var1, var2);*

*applyOppDamage(var1, var2);*

*}*

*}*

*protected String describe() { return (new String(Mes.use) + " Bone Rush"); }*

*}*

*BraveBird.java*

*package Moves;*

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

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

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

*import Lib.Mes;*

*public class BraveBird extends PhysicalMove {*

*public BraveBird() { super(Type.FLYING, 120, 100); }*

*protected void applySelfDamage(Pokemon att, double damage) {*

*super.applySelfDamage(att, (damage / 3));*

*}*

*protected String describe() { return (new String(Mes.use) + " Brave Bird"); }*

*}*

*DoubleHit.java*

*package Moves;*

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

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

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

*import Lib.Mes;*

*public class DoubleHit extends PhysicalMove {*

*public DoubleHit() { super(Type.NORMAL, 35, 90); }*

*protected void applyOppDamage(Pokemon var1, double var2) {*

*super.applyOppDamage(var1, var2);*

*super.applyOppDamage(var1, var2);*

*}*

*protected String describe() { return (new String(Mes.use) + " Double Hit"); }*

*}*

*DragonBreath.java*

*package Moves;*

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

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

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

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

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

*import Lib.Mes;*

*public class DragonBreath extends SpecialMove {*

*public DragonBreath() { super(Type.DRAGON, 60, 100); }*

*protected void applyOppEffects(Pokemon var1) {*

*Effect var2 = (new Effect()).chance(0.3).condition(Status.PARALYZE);*

*var1.setCondition(var2);*

*}*

*protected String describe() { return (new String(Mes.use) + " Dragon Breath"); }*

*}*

*DragonPulse.java*

*package Moves;*

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

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

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

*import Lib.Mes;*

*public class DragonPulse extends SpecialMove {*

*public DragonPulse() { super(Type.DRAGON, 85, 100); }*

*protected void applyOppDamage(Pokemon var1, double var2) {*

*super.applyOppDamage(var1, var2);*

*}*

*protected String describe() { return (new String(Mes.use) + " Dragon Pulse"); }*

*}*

*Facade.java*

*package Moves;*

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

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

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

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

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

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

*import Lib.Mes;*

*public class Facade extends PhysicalMove {*

*public Facade() { super(Type.NORMAL, 70, 100); }*

*protected void applySelfEffects(Pokemon var1) {*

*if (var1.getCondition() == Status.BURN || var1.getCondition() == Status.PARALYZE || var1.getCondition() == Status.POISON) {*

*Effect var2 = (new Effect()).turns(0).stat(Stat.ATTACK, (int)(var1.getStat(Stat.ATTACK) \* 2));*

*var1.addEffect(var2);*

*}*

*}*

*protected String describe() { return (new String(Mes.use) + " Facade"); }*

*}*

*Leer.java*

*package Moves;*

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

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

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

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

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

*import Lib.Mes;*

*public class Leer extends StatusMove {*

*public Leer() { super(Type.NORMAL, 0, 100); }*

*protected void applyOppEffects(Pokemon var1) {*

*Effect var2 = new Effect().turns(0).stat(Stat.DEFENSE, -1);*

*var1.addEffect(var2);*

*}*

*protected String describe() { return (new String(Mes.use) + " Leer"); }*

*}*

*Roost.java*

*package Moves;*

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

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

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

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

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

*import Lib.Mes;*

*public class Roost extends StatusMove {*

*public Roost() { super(Type.FLYING, 0, 0); }*

*protected void applySelfEffects(Pokemon var1) {*

*Effect var2 = (new Effect()).turns(0).stat(Stat.HP, (int)(var1.getStat(Stat.HP) \* 1.5));*

*var1.addEffect(var2);*

*}*

*protected String describe() { return (new String(Mes.use) + " Roost"); }*

*}*

*ShadowBall.java*

*package Moves;*

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

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

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

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

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

*import Lib.Mes;*

*public class ShadowBall extends SpecialMove {*

*public ShadowBall() { super(Type.GHOST, 80, 100); }*

*protected void applyOppEffects(Pokemon var1) {*

*Effect var2 = new Effect().chance(0.2).turns(0).stat(Stat.SPECIAL\_DEFENSE, -1);*

*var1.addEffect(var2);*

*}*

*protected String describe() { return (new String(Mes.use) + " Shadow Ball"); }*

*}*

*Slash.java*

*package Moves;*

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

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

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

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

*import Lib.Mes;*

*public class Slash extends PhysicalMove {*

*public Slash() { super(Type.NORMAL, 70, 100); }*

*protected double calcCriticalHit(Pokemon var1, Pokemon var2) {*

*if (var1.getStat(Stat.SPEED) / 256.0 > Math.random()) {*

*System.out.println(new String(Mes.crit));*

*return 2.0;*

*} else {*

*return 1.0;*

*}*

*}*

*protected String describe() { return (new String(Mes.use) + " Slash"); }*

*}*

*StoneEdge.java*

*package Moves;*

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

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

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

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

*import Lib.Mes;*

*public class StoneEdge extends PhysicalMove {*

*public StoneEdge() { super(Type.ROCK, 100, 80); }*

*protected double calcCriticalHit(Pokemon var1, Pokemon var2) {*

*if (var1.getStat(Stat.SPEED) / 256.0 > Math.random()) {*

*System.out.println(new String(Mes.crit));*

*return 2.0;*

*} else {*

*return 1.0;*

*}*

*}*

*protected String describe() { return (new String(Mes.use) + " Stone Edge"); }*

*}*

*Mes.java*

*package Lib;*

*public class Mes {*

*public static final char[] use = {1080, 1089, 1087, 1086, 1083, 1100, 1079, 1091, 1077, 1090};*

*public static final char[] crit = {1050, 1088, 1080, 1090, 1080, 1095, 1077, 1089, 1082, 1080, 1081, 32, 1091, 1076, 1072, 1088, 33};*

*public static final char[] Vafla = {1042, 1072, 1092, 1083, 1103};*

*public static final char[] Tolya = {1058, 1086, 1083, 1103};*

*public static final char[] Mortis = {1052, 1086, 1088, 1090, 1080, 1089};*

*public static final char[] Fermer = {1060, 1077, 1088, 1084, 1077, 1088};*

*}*

Результат работы программы

Hydreigon Фермер из команды желтых вступает в бой!

Deino Вафля из команды полосатых вступает в бой!

Hydreigon Фермер использует Stone Edge.

Deino Вафля теряет 6 здоровья.

Deino Вафля использует Dragon Breath.

Hydreigon Фермер теряет 14 здоровья.

Hydreigon Фермер парализован

Hydreigon Фермер теряет сознание.

Vullaby Voryga из команды желтых вступает в бой!

Vullaby Voryga промахивается

Deino Вафля использует Dragon Breath.

Vullaby Voryga теряет 5 здоровья.

Vullaby Voryga использует Shadow Ball.

Deino Вафля теряет 2 здоровья.

Deino Вафля использует Dragon Breath.

Vullaby Voryga теряет 8 здоровья.

Vullaby Voryga парализован

Vullaby Voryga теряет сознание.

Mandibuzz Pikachu из команды желтых вступает в бой!

Mandibuzz Pikachu промахивается

Deino Вафля использует Dragon Breath.

Mandibuzz Pikachu теряет 7 здоровья.

Mandibuzz Pikachu промахивается

Deino Вафля использует Dragon Pulse.

Mandibuzz Pikachu теряет 7 здоровья.

Mandibuzz Pikachu теряет сознание.

В команде желтых не осталось покемонов.

Команда полосатых побеждает в этом бою!

Вывод

Я вспомнил как нужно организовывать программы с множеством классов. Научился использовать jar файл как библиотеку, компилировать и запускать все это через консоль. Понял насколько важна и полезна бывает документация. Узнал как в java реализуется связь с классом-родителем.