«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет программной инженерии и компьютерной техники  
Направление подготовки 09.03.04 Программная инженерия  
Дисциплина «Программирование»

Отчет

По лабораторной работе №2

Вариант 290002

Студент:

*Ильин Н. С.*

*Р3110 поток 2.9*

Преподаватель:

*Наумова Н. А.*

Санкт-Петербург, 2023 г.

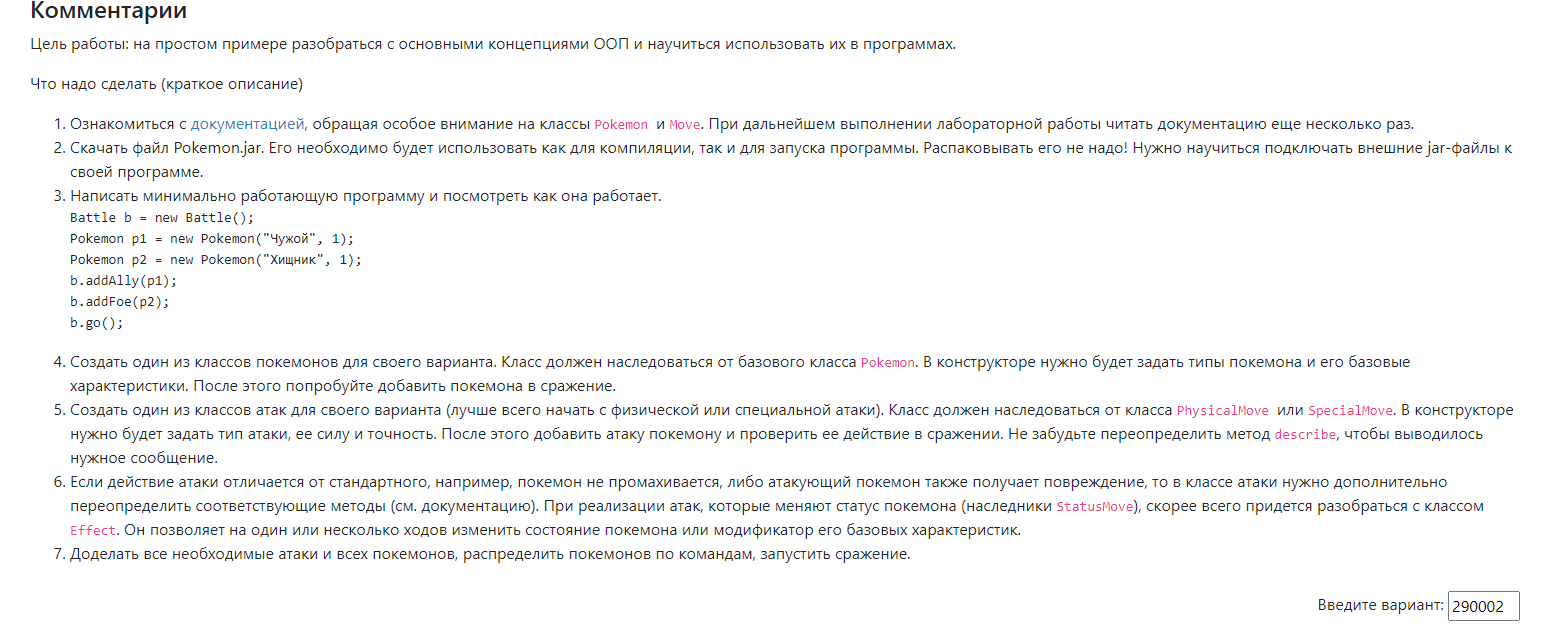
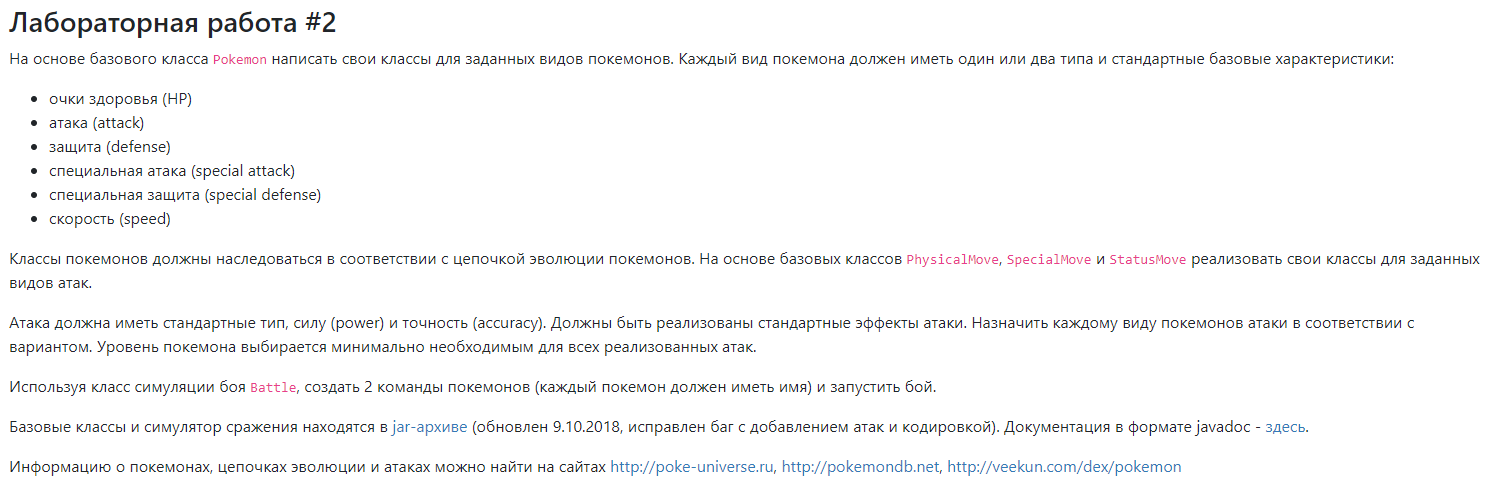
**Оглавление**

[Задание: 3](#_Toc145408419)

[Выполнение работы: 4](#_Toc145408420)

[Выводы: 12](#_Toc145408421)

# **Задание:**





# **Выполнение работы:**

Исходный код:

#### src/Main.java

**import mypokemons.\*;**

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

**public class Main {**

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

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

**b.addAlly(new Sealeo("Силео", 1));**

**b.addAlly(new Skarmory("Скармори", 1));**

**b.addAlly(new Houndour("Хандур", 1));**

**b.addFoe(new Oddish("Оддиш", 1));**

**b.addFoe(new Gloom("Глум", 1));**

**b.addFoe(new Totodile("Тотодил", 1));**

**b.go();**

**}**

**}**

#### src/utils/Utils.java

**package utils;**

**public class Utils {**

**public static boolean chance(double chance) {**

**return Math.random() <= chance;**

**}**

**}**

#### src/mypokemons/Gloom.java

**package mypokemons;**

**import mymoves.\*;**

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

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

**public class Gloom extends Pokemon {**

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

**super(name, level);**

**super.setType(Type.GRASS, Type.POISON);**

**super.setStats(60,65,70,85,75,40);**

**super.setMove(new LightScreen(), new HydroPump(), new Charge());**

**}**

**}**

#### src/mypokemons/Houndour.java

**package mypokemons;**

**import mymoves.\*;**

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

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

**public class Houndour extends Pokemon {**

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

**super(name, level);**

**super.setType(Type.DARK, Type.FIRE);**

**super.setStats(45,60, 30,80,50,65);**

**super.setMove(new HydroPump(), new ThunderShock(), new Endeavor(), new Bite());**

**}**

**}**

#### src/mypokemons/Oddish.java

**package mypokemons;**

**import mymoves.\*;**

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

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

**public class Oddish extends Pokemon {**

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

**super(name, level);**

**super.setType(Type.GRASS, Type.POISON);**

**super.setStats(45,50,55,75,65,30);**

**super.setMove(new LightScreen(), new HydroPump());**

**}**

**}**

#### src/mypokemons/Sealeo.java

**package mypokemons;**

**import mymoves.\*;**

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

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

**public class Sealeo extends Pokemon {**

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

**super(name, level);**

**super.setType(Type.ICE, Type.GHOST);**

**super.setStats(90, 60, 70, 75, 70, 45);**

**super.setMove(new DoubleEdge(), new Endeavor(), new Substitute(), new TakeDown());**

**}**

**}**

­­­

#### src/mypokemons/Skarmory.java

**package mypokemons;**

**import mymoves.\*;**

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

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

**public class Skarmory extends Pokemon {**

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

**super(name, level);**

**super.setType(Type.STEEL, Type.FLYING);**

**super.setStats(65, 80, 140, 40, 70, 70);**

**super.setMove(new HydroPump(), new ThunderShock(), new Endeavor());**

**}**

**}**

#### src/mypokemons/Totodile.java

**package mypokemons;**

**import mymoves.\*;**

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

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

**public class Totodile extends Pokemon {**

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

**super(name, level);**

**super.setType(Type.WATER);**

**super.setStats(50,65,64,44,48,43);**

**super.setMove(new LightScreen(), new HydroPump(), new Charge(), new Meditate());**

**}**

**}**

#### src/mymoves/Bite.java

**package mymoves;**

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

**import utils.Utils;**

**public class Bite extends PhysicalMove {**

**public Bite() {**

**super(Type.DARK, 60, 100);**

**}**

**@Override**

**protected void applyOppDamage(Pokemon def, double damage) {**

**super.applyOppDamage(def, damage);**

**if(Utils.chance(0.3)) {**

**Effect.flinch(def);**

**}**

**}**

**@Override**

**protected String describe(){**

**return "использует Bite";**

**}**

**}**

#### src/mymoves/Charge.java

**package mymoves;**

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

**import utils.Utils;**

**public class Charge extends PhysicalMove {**

**public Charge() {**

**super(Type.ELECTRIC, 0, 100);**

**}**

**@Override**

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

**p.addEffect(new Effect().turns(5).stat(Stat.SPECIAL\_DEFENSE, 1));**

**}**

**@Override**

**protected String describe(){**

**return "использует Charge";**

**}**

**}**

#### src/mymoves/DoubleEdge.java

**package mymoves;**

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

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

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

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

**public class DoubleEdge extends PhysicalMove{**

**public DoubleEdge(){**

**super(Type.NORMAL,120,100);**

**}**

**@Override**

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

**p.setMod(Stat.HP,(int)(damage/3.0));**

**}**

**@Override**

**protected String describe(){**

**return "использовал Double-Edge";**

**}**

**}**

#### src/mymoves/Endeavor.java

**package mymoves;**

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

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

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

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

**public class Endeavor extends PhysicalMove {**

**public Endeavor() {**

**super(Type.NORMAL,0,100);**

**}**

**@Override**

**protected double calcBaseDamage(Pokemon att, Pokemon def) {**

**double attHP = att.getHP();**

**double defHP = def.getHP();**

**if (defHP > attHP) {**

**def.setMod(Stat.HP, (int)(defHP-attHP));**

**}**

**return 0.0;**

**}**

**@Override**

**protected String describe(){**

**return "использует Endeavor";**

**}**

**}**

#### src/mymoves/HydroPump.java

**package mymoves;**

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

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

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

**public class HydroPump extends PhysicalMove {**

**public HydroPump() {**

**super(Type.WATER, 110, 80);**

**}**

**@Override**

**protected String describe(){**

**return "использует Hydro Pump";**

**}**

**}**

#### src/mymoves/LightScreen.java

**package mymoves;**

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

**import utils.Utils;**

**public class LightScreen extends PhysicalMove {**

**public LightScreen() {**

**super(Type.PSYCHIC, 0, 100);**

**}**

**@Override**

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

**p.addEffect(new Effect().turns(5).stat(Stat.SPECIAL\_DEFENSE, (int)p.getStat(Stat.SPECIAL\_DEFENSE)/2));**

**}**

**@Override**

**protected String describe(){**

**return "использует Light Screen";**

**}**

**}**

#### src/mymoves/Meditate.java

**package mymoves;**

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

**public class Meditate extends PhysicalMove {**

**public Meditate() {**

**super(Type.PSYCHIC, 0, 100);**

**}**

**@Override**

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

**p.addEffect(new Effect().turns(5).stat(Stat.ATTACK, 1));**

**}**

**@Override**

**protected String describe(){**

**return "использует Meditate";**

**}**

**}**

#### src/mymoves/Substitute.java

**package mymoves;**

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

**public class Substitute extends PhysicalMove {**

**public Substitute(){**

**super(Type.NORMAL,0,100);**

**}**

**@Override**

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

**double currSelfHp = p.getHP();**

**p.setMod(Stat.HP,(int)(currSelfHp\*(-2)));**

**}**

**@Override**

**protected String describe(){**

**return "использовал Substitute";**

**}**

**}**

#### src/mymoves/TakeDown.java

**package mymoves;**

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

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

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

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

**public class TakeDown extends PhysicalMove {**

**public TakeDown(){**

**super(Type.NORMAL,90,85);**

**}**

**@Override**

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

**p.setMod(Stat.HP,(int)(damage/4));**

**}**

**@Override**

**protected String describe(){**

**return "использовал Double-Edge";**

**}**

**}**

#### src/mymoves/ThunderShock.java

**package mymoves;**

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

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

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

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

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

**public class ThunderShock extends SpecialMove{**

**public ThunderShock () {**

**super(Type.ELECTRIC, 40, 100);**

**}**

**@Override**

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

**p.addEffect(new Effect().chance(0.1).condition(Status.PARALYZE));**

**}**

**@Override**

**protected String describe (){**

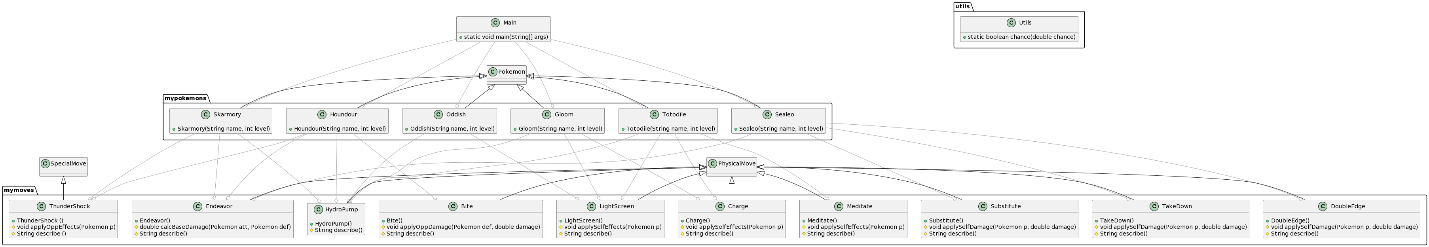
**return "использует Thunder Shock";**

**}**

**}**

Диаграмма классов:

[(svg)](https://www.plantuml.com/plantuml/svg/hLNDRjim3BxxAGJquc9Ddg7f4go71ItWx5Jqe5YCBKIM34dE5Wnxzzck5GYiF2v0BWKQyllX9qf-T9vRtpKgAnLtZcsvrEntziLvxcN9JaOArd2fyy9RgQktTyPjvUwpFzdOkLW8yrE_tQrMg_U5OSsvDKTeZ7RB0hW2CtL5JQZho6rZx1dt8R9dqsbXEalHlGWXNKsGVLF6D9JzdV56I0NPb4ldfNBB7yFVA9wzCGguPcNDTGbvRt6lOFflo2YWiJg3NnwqSEnr_7bCV2nCCJFD6tXWKdkcu0HguC8SdsDep0dSyhkiQb-K5a3_4nD1di_2cjUkQVzRv6lDRHN2IPrKQi9G8uR4J8JdD2ODsjLz7MnHc_98QaMuJO_Y9sElyFSaVPGFsj1KEXyLCfkOXTu00fIACtZ6Lpdb-QD2UuvOPavcFLwTZQWWXQ5F9Or08aMtTrxwpbyXikD7MFULH2xXyKeb4sf4DeY9EfmqXjxr98v8DJ5mU1HJ0uSQDJ1YIdX76qoXKZcH8FoIfG5fn1Q4z3o-tPSR7oLJdvqikThsfI6UePZVPtVZ1vMthJg_jEsQDxo2FAGeuF30fc-864izO8fJW2kjt4CUVLRG3BFQOncf5w0EcyC1Ikyk-YqjX-xgh6HeIScYgTihPqPhEciXD0m69e6-MZvn1yW1z_w1HOT0Ea3RDclXqf7Jb7ZlPqcZdjiTQTZfMTNGS1D9z1ZFYbvQRhz5w7MVjH0QRcuWyKfVJ1Gjb3Bo4FTyUaJe_85DqeyzUzUex2y0)



Вывод программы:

Sealeo Силео из команды фиолетовых вступает в бой!

Oddish Оддиш из команды желтых вступает в бой!

Sealeo Силео использовал Double-Edge.

Oddish Оддиш теряет 5 здоровья.

Sealeo Силео теряет 1 здоровья.

Oddish Оддиш использует Hydro Pump.

Sealeo Силео теряет 5 здоровья.

Sealeo Силео использовал Double-Edge.

Oddish Оддиш теряет 5 здоровья.

Sealeo Силео теряет 1 здоровья.

Oddish Оддиш использует Light Screen.

Sealeo Силео теряет 2 здоровья.

Sealeo Силео использовал Substitute.

Oddish Оддиш теряет 2 здоровья.

Sealeo Силео восстанавливает 7 здоровья.

Oddish Оддиш использует Light Screen.

Sealeo Силео теряет 3 здоровья.

Sealeo Силео использовал Substitute.

Oddish Оддиш теряет 3 здоровья.

Sealeo Силео восстанавливает 15 здоровья.

Oddish Оддиш теряет сознание.

Gloom Глум из команды желтых вступает в бой!

Sealeo Силео использовал Substitute.

Gloom Глум теряет 3 здоровья.

Sealeo Силео восстанавливает 45 здоровья.

Gloom Глум использует Charge.

Sealeo Силео теряет 2 здоровья.

Sealeo Силео использует Endeavor.

Gloom Глум теряет 2 здоровья.

Gloom Глум использует Hydro Pump.

Sealeo Силео теряет 4 здоровья.

Sealeo Силео использует Endeavor.

Gloom Глум теряет 3 здоровья.

Gloom Глум использует Light Screen.

Sealeo Силео теряет 2 здоровья.

Sealeo Силео использовал Substitute.

Gloom Глум теряет 3 здоровья.

Sealeo Силео восстанавливает 119 здоровья.

Gloom Глум использует Charge.

Sealeo Силео теряет 2 здоровья.

Sealeo Силео использовал Substitute.

Gloom Глум теряет 4 здоровья.

Sealeo Силео восстанавливает 353 здоровья.

Gloom Глум теряет сознание.

Totodile Тотодил из команды желтых вступает в бой!

Sealeo Силео использовал Double-Edge.

Totodile Тотодил теряет 5 здоровья.

Sealeo Силео теряет 1 здоровья.

Totodile Тотодил использует Light Screen.

Sealeo Силео теряет 2 здоровья.

Sealeo Силео использовал Double-Edge.

Totodile Тотодил теряет 5 здоровья.

Sealeo Силео теряет 1 здоровья.

Totodile Тотодил использует Meditate.

Sealeo Силео теряет 3 здоровья.

Sealeo Силео использовал Double-Edge.

Totodile Тотодил теряет 4 здоровья.

Sealeo Силео теряет 1 здоровья.

Totodile Тотодил теряет сознание.

В команде желтых не осталось покемонов.

Команда фиолетовых побеждает в этом бою!

# **Выводы**:

В результате проделанной работы, я познакомился с основами ООП, научился подключать внешние библиотеки, изучил работу методов, классов и объектов.