Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники



Лабораторная работа №2   
по программированию

Вариант: 331625

Группа: P3114

Студент: Лагус

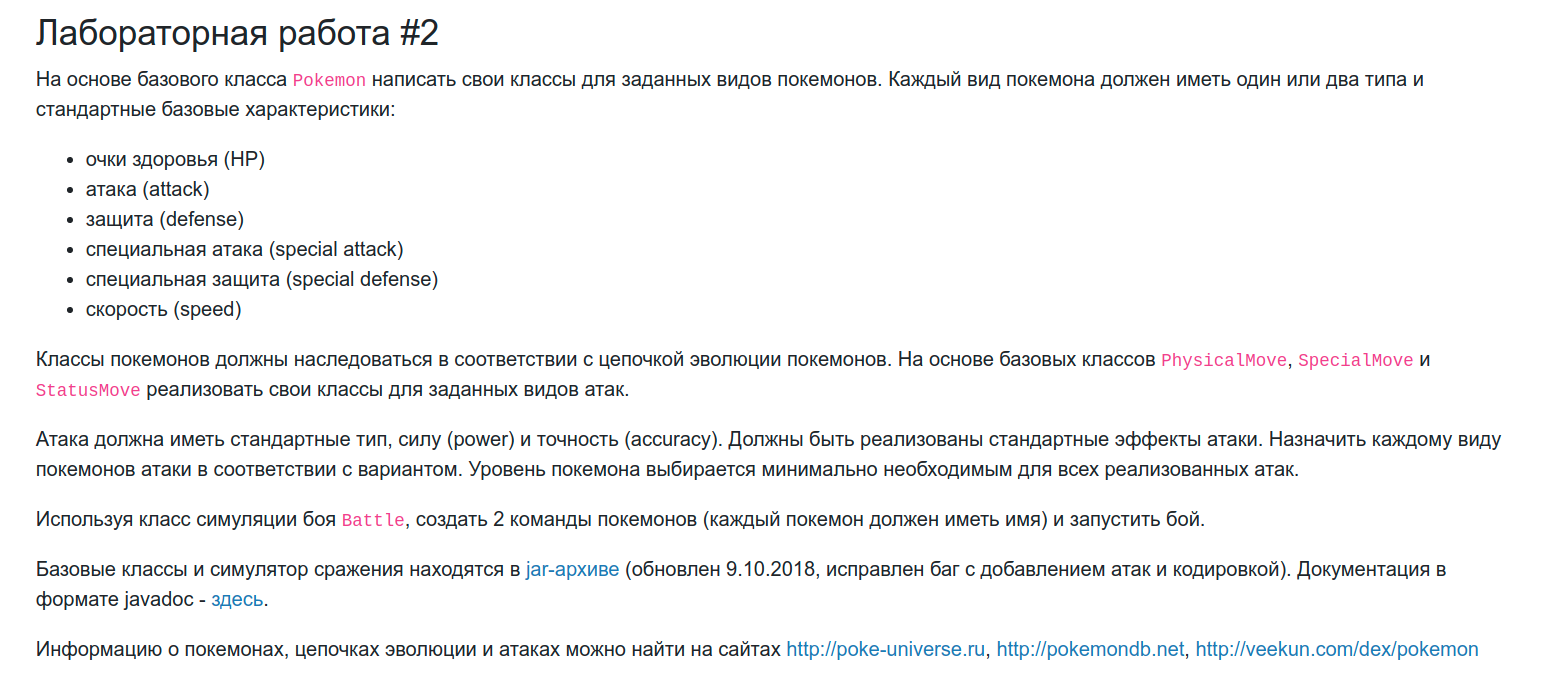
Максим Сергеевич

Преподаватель: Письмак Алексей Евгеньевич

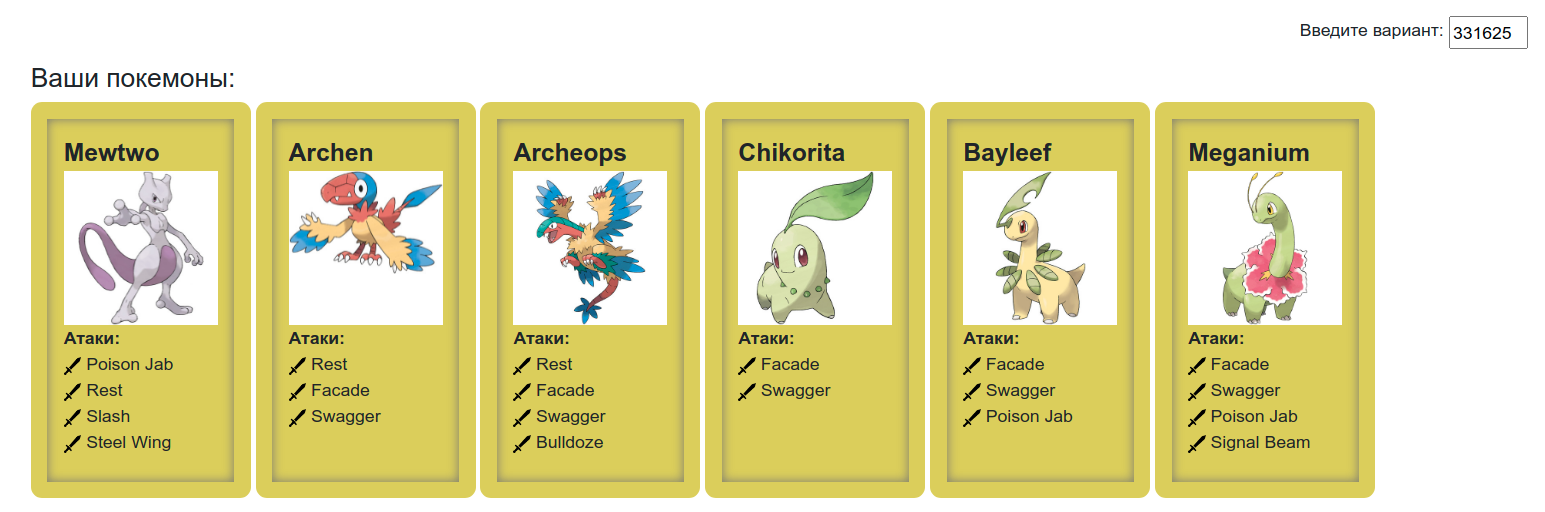
г. Санкт-Петербург

Сентябрь, 2021

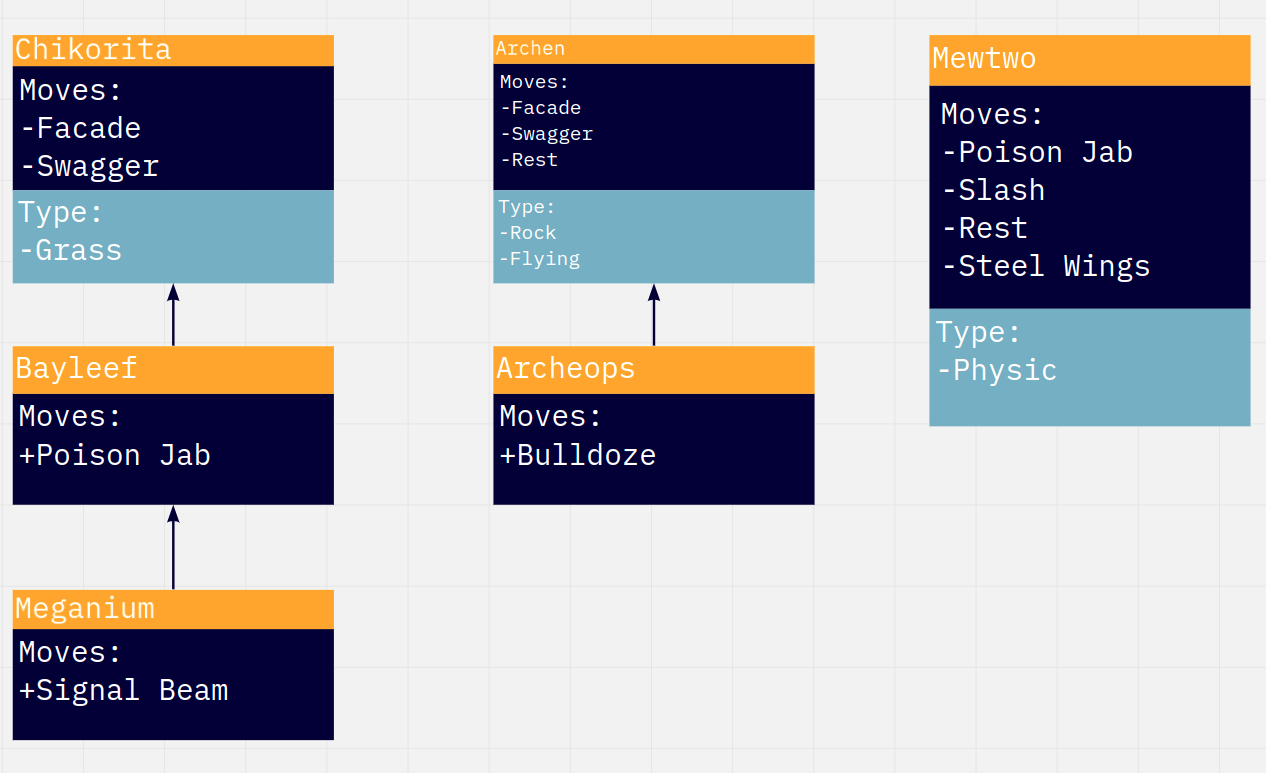
**Текст задания**



**Мои покемоны согласно варианту**



**Диаграмма классов реализованной объектной модели.**



**Исходный код программы**

**Moves:**

**Bulldoze.java**

package Moves;

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 Bulldoze extends PhysicalMove {

public Bulldoze() {

super(Type.GROUND, 60, 100);

}

@Override

protected void applyOppEffects(Pokemon pokemon) {

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

}

@Override

protected String describe() {

return "наносит урон и понижает скорость";

}

}

**Facade.java**

package Moves;

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

public class Facade extends PhysicalMove {

private boolean isCritical;

public Facade() {

super(Type.NORMAL, 70, 100);

isCritical = false;

}

@Override

protected double calcCriticalHit(Pokemon pokemon, Pokemon pokemon1) {

if (pokemon.getCondition() == Status.BURN

| pokemon.getCondition() == Status.POISON

| pokemon.getCondition() == Status.PARALYZE) {

isCritical = true;

return 2.0;

} else return super.calcCriticalHit(pokemon, pokemon1);

}

@Override

protected String describe() {

if (isCritical)

return "наносит двойной урон";

return "наносит урон";

}

}

**PoisonJab.java**

package Moves;

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

public class PoisonJab extends PhysicalMove {

private boolean isPoisoned;

public PoisonJab() {

super(Type.POISON, 80, 100);

isPoisoned = false;

}

@Override

protected void applyOppEffects(Pokemon pokemon) {

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

isPoisoned = true;

Effect.poison(pokemon);

}

}

@Override

protected String describe() {

if (isPoisoned) {

return "наносит урон и отравляет";

}

return "наносит урон";

}

}

**Rest.java**

package Moves;

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

public class Rest extends StatusMove {

public Rest() {

super(Type.PSYCHIC, 0, 0);

}

@Override

protected void applySelfEffects(Pokemon pokemon) {

pokemon.restore();

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

pokemon.setCondition(effect);

pokemon.setMod(Stat.HP, 6);

}

@Override

protected String describe() {

return "использовавший данную способность покемон отдыхает";

}

}

**SignalBeam.java**

package Moves;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.SpecialMove;

import ru.ifmo.se.pokemon.Type;

public class SignalBeam extends SpecialMove {

private boolean isConfused ;

public SignalBeam() {

super(Type.BUG, 75, 100);

isConfused = false;

}

@Override

protected void applyOppEffects(Pokemon pokemon) {

if (Math.random() < 0.1) {

isConfused = true;

pokemon.confuse();

}

}

@Override

protected String describe() {

if (isConfused)

return "атакует и вводит врага в замешательство";

return "атакует";

}

}

**Slash.java**

package Moves;

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

public class Slash extends PhysicalMove {

public Slash() {

super(Type.NORMAL, 70, 100);

}

@Override

protected double calcCriticalHit(Pokemon pokemon, Pokemon pokemon1) {

//We don't have ability to implement this side effect

return super.calcCriticalHit(pokemon, pokemon1);

}

@Override

protected String describe() {

return "наносит урон";

}

}

**SteelWing.java**

package Moves;

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

public class SteelWing extends PhysicalMove {

private boolean isLowered;

public SteelWing() {

super(Type.STEEL, 70, 90);

isLowered = false;

}

@Override

protected void applySelfEffects(Pokemon pokemon) {

if (Math.random() <= 0.1) {

isLowered = true;

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

}

}

@Override

protected String describe() {

if (isLowered) {

return "повышает защиту и наносит урон";

}

return "наносит урон";

}

}

**Swagger.java**

package Moves;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Stat;

import ru.ifmo.se.pokemon.StatusMove;

import ru.ifmo.se.pokemon.Type;

public class Swagger extends StatusMove {

public Swagger() {

super(Type.NORMAL, 0, 85);

}

@Override

protected void applyOppEffects(Pokemon pokemon) {

pokemon.confuse();

pokemon.setMod(Stat.ATTACK, 2);

}

@Override

protected String describe() {

return "сбивает противника с толку и повышает его атаку";

}

}

**Pokemons:**

**Archen.java**

package Pokemons;

import Moves.Facade;

import Moves.Rest;

import Moves.Swagger;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

public class Archen extends Pokemon {

public Archen(String name, int lvl) {

super(name, lvl);

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

setType(Type.ROCK,Type.FLYING);

setMove(new Rest(),new Facade(),new Swagger());

}

}

**Archeops.java**

package Pokemons;

import Moves.Bulldoze;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

public class Archeops extends Archen {

public Archeops(String name, int lvl) {

super(name, lvl);

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

addMove(new Bulldoze());

}

}

**Bayleef.java**

package Pokemons;

import Moves.PoisonJab;

import ru.ifmo.se.pokemon.Pokemon;

public class Bayleef extends Chikorita {

public Bayleef(String name, int lvl) {

super(name, lvl);

setStats(60,62,80,63,80,60);

addMove(new PoisonJab());

}

}

**Chikorita.java**

package Pokemons;

import Moves.Facade;

import Moves.Swagger;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

public class Chikorita extends Pokemon {

public Chikorita(String name, int lvl) {

super(name, lvl);

setType(Type.GRASS);

setStats(45,49,65,49,65,45);

setMove(new Facade(),new Swagger());

}

}

**Meganium.java**

package Pokemons;

import Moves.SignalBeam;

public class Meganium extends Bayleef{

public Meganium(String name, int lvl) {

super(name, lvl);

setStats(80,82,100,83,100,80);

addMove(new SignalBeam());

}

}

**Mewtwo.java**

package Pokemons;

import Moves.\*;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

public class Mewtwo extends Pokemon {

public Mewtwo(String name, int lvl) {

super(name, lvl);

setStats(106,110,90,154,90,130);

setType(Type.PSYCHIC);

setMove(new PoisonJab(),new Rest(),new Slash(), new SteelWing());

}

}

**Main.java**

import Pokemons.\*;

import ru.ifmo.se.pokemon.Battle;

public class Main {

public static void main(String[] args) {

Battle battle = new Battle();

battle.addAlly(new Archen("Kvothe",20));

battle.addAlly(new Mewtwo("Kaladin",25));

battle.addAlly(new Bayleef("Vin",20));

battle.addFoe(new Archeops("Shrike",20));

battle.addFoe(new Chikorita("Ged",20));

battle.addFoe(new Meganium("Fitz",20));

battle.go();

}

}

**Результат работы программы**

Archen Kvothe from the team Yellow enters the battle!

Archeops Shrike from the team Red enters the battle!

Archeops Shrike наносит урон.

Archen Kvothe loses 13 hit points.

Archen Kvothe misses

Archeops Shrike сбивает противника с толку и повышает его атаку.

Archen Kvothe increases attack.

Archen Kvothe misses

Archeops Shrike наносит урон и понижает скорость.

Archen Kvothe loses 1 hit points.

Archen Kvothe isn't affected by GROUND

Archen Kvothe hits himself in confusion.

Archen Kvothe loses 18 hit points.

Archeops Shrike наносит урон и понижает скорость.

Archen Kvothe loses 1 hit points.

Archen Kvothe isn't affected by GROUND

Archen Kvothe наносит урон.

Archeops Shrike loses 10 hit points.

Archeops Shrike сбивает противника с толку и повышает его атаку.

Archen Kvothe increases attack.

Archen Kvothe hits himself in confusion.

Archen Kvothe loses 19 hit points.

Archeops Shrike сбивает противника с толку и повышает его атаку.

Archen Kvothe increases attack.

Archen Kvothe наносит урон.

Archeops Shrike loses 25 hit points.

Archeops Shrike наносит урон и понижает скорость.

Archen Kvothe loses 1 hit points.

Archen Kvothe isn't affected by GROUND

Archen Kvothe наносит урон.

Archeops Shrike loses 20 hit points.

Archeops Shrike сбивает противника с толку и повышает его атаку.

Archen Kvothe increases attack.

Archen Kvothe hits himself in confusion.

Archen Kvothe loses 22 hit points.

Archen Kvothe faints.

Mewtwo Kaladin from the team Yellow enters the battle!

Mewtwo Kaladin misses

Archeops Shrike наносит урон.

Mewtwo Kaladin loses 13 hit points.

Mewtwo Kaladin наносит урон.

Archeops Shrike loses 7 hit points.

Archeops Shrike is poisoned

Archeops Shrike наносит урон и понижает скорость.

Mewtwo Kaladin loses 9 hit points.

Mewtwo Kaladin decreases speed.

Archeops Shrike loses 7 hit points.

Chikorita Ged from the team Red enters the battle!

Mewtwo Kaladin misses

Chikorita Ged сбивает противника с толку и повышает его атаку.

Mewtwo Kaladin increases attack.

Mewtwo Kaladin misses

Chikorita Ged наносит урон.

Mewtwo Kaladin loses 8 hit points.

Mewtwo Kaladin наносит урон.

Chikorita Ged loses 30 hit points.

Chikorita Ged сбивает противника с толку и повышает его атаку.

Mewtwo Kaladin increases attack.

Mewtwo Kaladin наносит урон и отравляет.

Critical hit!

Chikorita Ged loses 167 hit points.

Chikorita Ged faints.

Meganium Fitz from the team Red enters the battle!

Mewtwo Kaladin misses

Meganium Fitz атакует.

Mewtwo Kaladin loses 22 hit points.

Mewtwo Kaladin наносит урон.

Meganium Fitz loses 33 hit points.

Meganium Fitz сбивает противника с толку и повышает его атаку.

Mewtwo Kaladin increases attack.

Mewtwo Kaladin наносит урон.

Meganium Fitz loses 51 hit points.

Meganium Fitz faints.

Team Red loses its last Pokemon.

The team Yellow wins the battle!

Process finished with exit code 0

**Вывод по работе**

Я познакомился с базовыми принципами ООП и их применением в Java.