Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 2**

по дисциплине

‘ПРОГРАММИРОВАНИЕ’

Вариант №ядалвариант

*Выполнил:*

Студент группы P3113

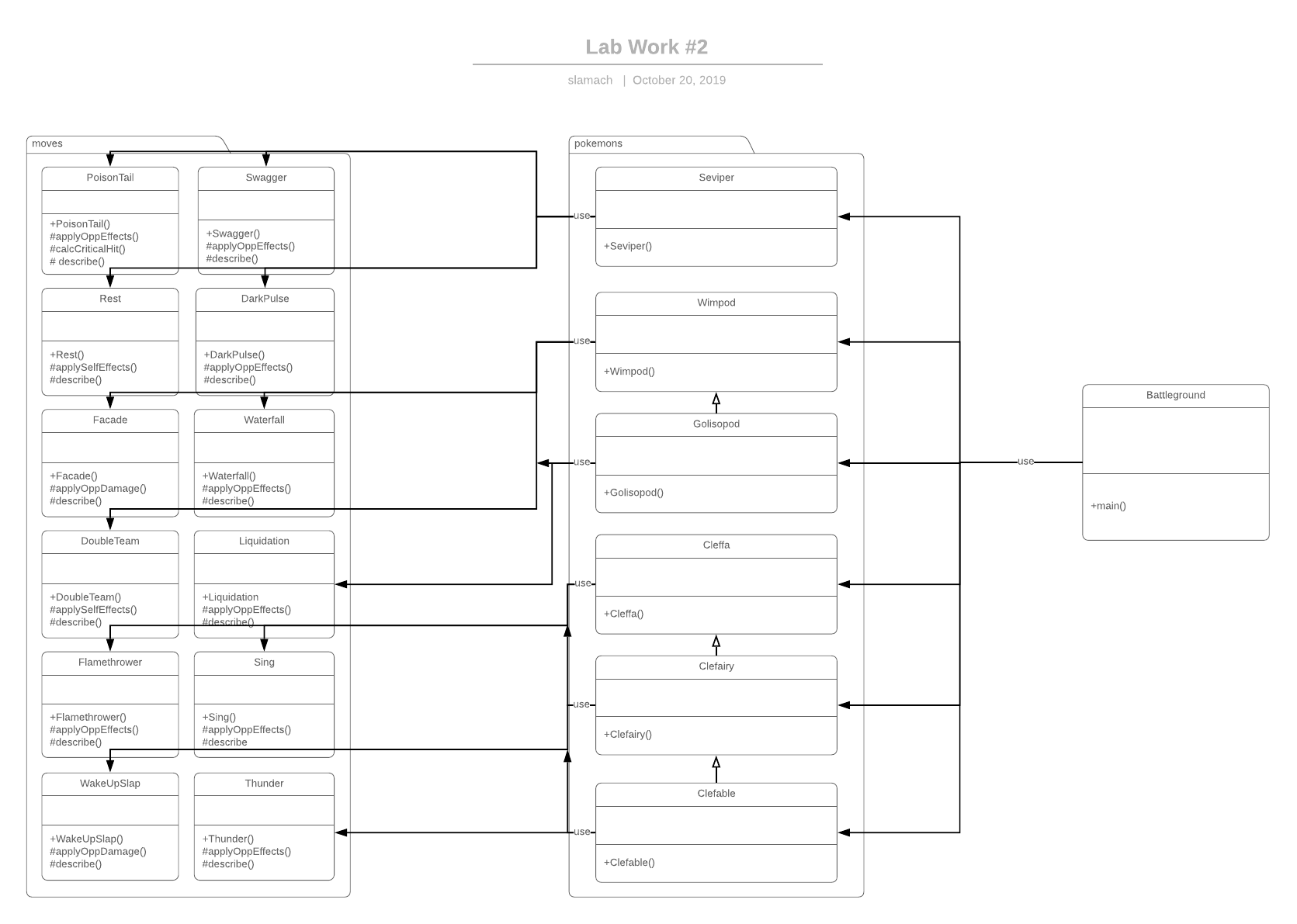
Свиридов Дмитрий Витальевич

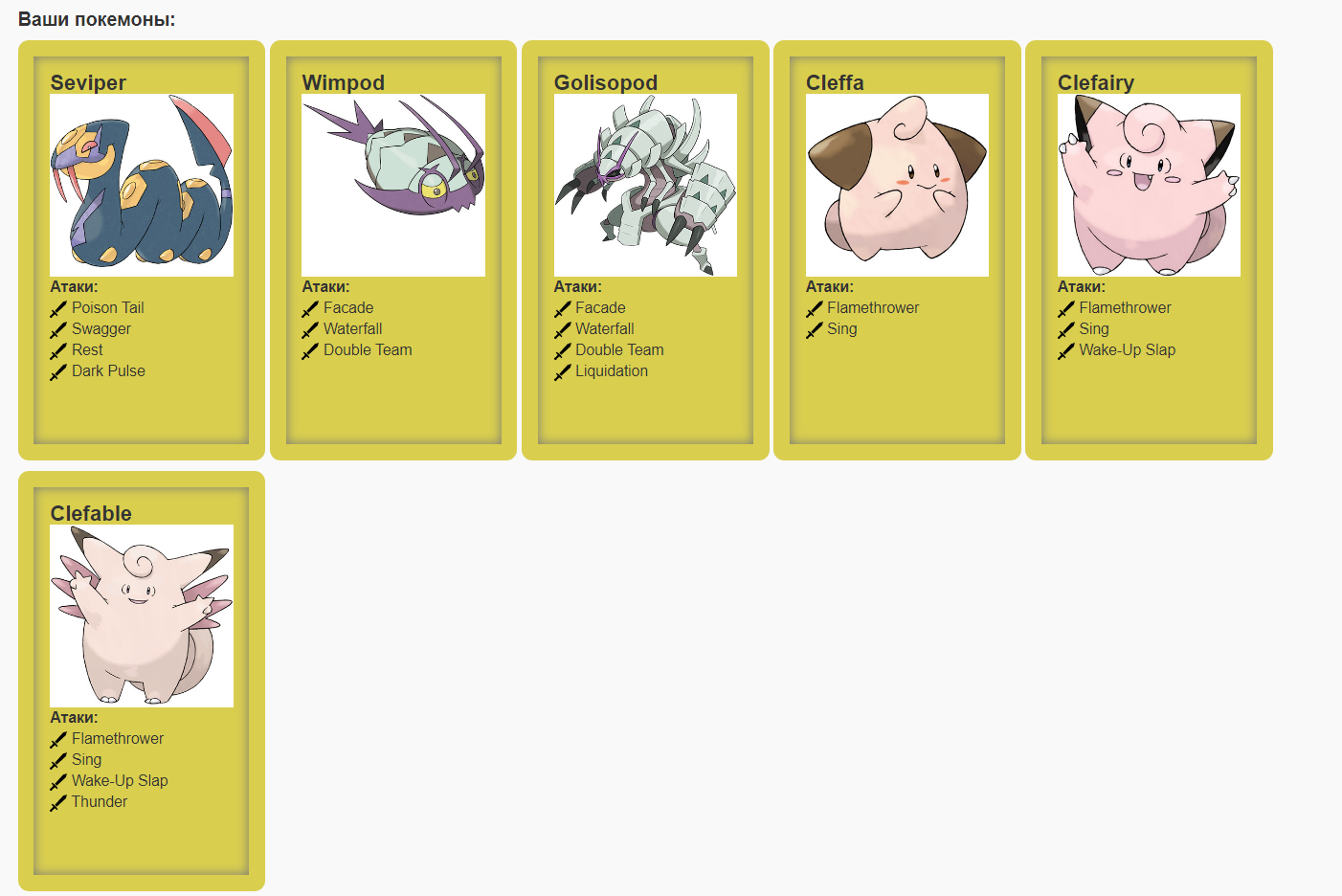
*Преподаватель:*

Письмак Алексей Евгеньевич



Санкт-Петербург, 2019

Диаграмма классов:

Покемоны:

Исходный код:

**Battleground.java**

import pokemons.\*;

import ru.ifmo.se.pokemon.Battle;

class Battleground {

    public static void main(String args[]) {

        Battle b = new Battle();

        Seviper p1 = new Seviper("Батя", 1);

        Wimpod p2 = new Wimpod("Брат", 2);

        Clefairy p3 = new Clefairy("Зять", 2);

        Golisopod p4 = new Golisopod("Дед", 1);

        Cleffa p5 = new Cleffa("Дядя", 3);

        Clefable p6 = new Clefable("Сосед", 1);

        b.addAlly(p1);

        b.addAlly(p2);

        b.addAlly(p3);

        b.addFoe(p4);

        b.addFoe(p5);

        b.addFoe(p6);

        b.go();

    }

}

**Seviper.java**

package pokemons;

import moves.\*;

import ru.ifmo.se.pokemon.\*;

public class Seviper extends Pokemon {

    public Seviper(String name, int level) {

        super(name, level);

        setStats(73, 100, 60, 100, 60, 65);

        setType(Type.POISON);

        setMove(new PoisonTail(), new Swagger(), new Rest(), new DarkPulse());

    }

}

**Wimpod.java**

package pokemons;

import moves.\*;

import ru.ifmo.se.pokemon.\*;

public class Wimpod extends Pokemon {

    public Wimpod(String name, int level) {

        super(name, level);

        setStats(25, 35, 40, 20, 30, 80);

        setType(Type.BUG, Type.WATER);

        setMove(new Facade(), new Waterfall(), new DoubleTeam());

    }

}

**Golisopod.java**

package pokemons;

import moves.\*;

public class Golisopod extends Wimpod {

    public Golisopod(String name, int level) {

        super(name, level);

        setStats(75, 125, 140, 60, 90, 40);

        addMove(new Liquidation());

    }

}

**Cleffa.java**

package pokemons;

import moves.\*;

import ru.ifmo.se.pokemon.\*;

public class Cleffa extends Pokemon {

    public Cleffa(String name, int level) {

        super(name, level);

        setStats(50, 25, 28, 45, 55, 15);

        setType(Type.FAIRY);

        setMove(new Flamethrower(), new Sing());

    }

}

**Clefairy.java**

package pokemons;

import moves.\*;

public class Clefairy extends Cleffa {

    public Clefairy(String name, int level) {

        super(name, level);

        setStats(70, 45, 48, 60, 65, 35);

        addMove(new WakeUpSlap());

    }

}

**Clefable.java**

package pokemons;

import moves.\*;

public class Clefable extends Clefairy {

    public Clefable(String name, int level) {

        super(name, level);

        setStats(95, 70, 73, 95, 90, 60);

        addMove(new Thunder());

    }

}

**PoisonTail.java**

package moves;

import ru.ifmo.se.pokemon.\*;

public class PoisonTail extends PhysicalMove {

    public PoisonTail() {

        super(Type.POISON, 50, 100);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        if (Math.random() < 0.1) {

            Effect.poison(p);

        }

    }

    @Override

    protected double calcCriticalHit(Pokemon att, Pokemon def) {

        if (Math.random() < (att.getStat(Stat.SPEED)/256.0)) {

            return 2.0;

        }

        else {

            return 1.0;

        }

    }

    @Override

    protected String describe() {

        return "использует Poison Tail";

    }

}

**Swagger.java**

package moves;

import ru.ifmo.se.pokemon.\*;

public class Swagger extends StatusMove {

    public Swagger() {

        super(Type.NORMAL, 0, 85);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        p.setMod(Stat.ATTACK, 2);

        Effect.confuse(p);

    }

    @Override

    protected String describe() {

        return "использует Swagger";

    }

}

**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 p) {

        Effect e1 = new Effect().turns(0).stat(Stat.HP, -99999);

        Effect e2 = new Effect().turns(2).condition(Status.SLEEP);

        p.addEffect(e1);

        p.setCondition(e2);

    }

    @Override

    protected String describe() {

        return "использует Rest";

    }

}

**DarkPulse.java**

package moves;

import ru.ifmo.se.pokemon.\*;

public class DarkPulse extends SpecialMove {

    public DarkPulse() {

        super(Type.DARK, 80, 100);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        if (Math.random() < 0.2) {

            Effect.flinch(p);

        }

    }

    @Override

    protected String describe() {

        return "использует Dark Pulse";

    }

}

**Facade.java**

package moves;

import ru.ifmo.se.pokemon.\*;

public class Facade extends PhysicalMove {

    public Facade() {

        super(Type.NORMAL, 70, 100);

    }

    @Override

    protected void applyOppDamage(Pokemon def, double damage) {

        Status p\_stat = def.getCondition();

        if (p\_stat.equals(Status.BURN) || p\_stat.equals(Status.POISON) || p\_stat.equals(Status.PARALYZE)) {

            super.applyOppDamage(def, damage\*2);

        }

        else {

            super.applyOppDamage(def, damage);

        }

    }

    @Override

    protected String describe() {

        return "использует Facade";

    }

}

**Waterfall.java**

package moves;

import ru.ifmo.se.pokemon.\*;

public class Waterfall extends PhysicalMove {

    public Waterfall() {

        super(Type.WATER, 80, 100);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        if (Math.random() < 0.2) {

            Effect.flinch(p);

        }

    }

    @Override

    protected String describe() {

        return "использует Waterfall";

    }

}

**DoubleTeam.java**

package moves;

import ru.ifmo.se.pokemon.\*;

public class DoubleTeam extends StatusMove {

    public DoubleTeam() {

        super(Type.NORMAL, 0, 0);

    }

    @Override

    protected void applySelfEffects(Pokemon p) {

        p.setMod(Stat.EVASION, 1);

    }

    @Override

    protected String describe() {

        return "использует Double Team";

    }

}

**Liquidation.java**

package moves;

import ru.ifmo.se.pokemon.\*;

public class Liquidation extends PhysicalMove {

    public Liquidation() {

        super(Type.WATER, 85, 100);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        if (Math.random() < 0.1) {

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

        }

    }

    @Override

    protected String describe() {

        return "использует Liquidation";

    }

}

**Flamethrower.java**

package moves;

import ru.ifmo.se.pokemon.\*;

public class Flamethrower extends SpecialMove {

    public Flamethrower() {

        super(Type.FIRE, 90, 100);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        if (Math.random() < 0.1) {

            Effect.burn(p);

        }

    }

    @Override

    protected String describe() {

        return "использует Flamethrower";

    }

}

**Sing.java**

package moves;

import ru.ifmo.se.pokemon.\*;

public class Sing extends StatusMove {

    public Sing() {

        super(Type.NORMAL, 0, 55);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        Effect.sleep(p);

    }

    @Override

    protected String describe() {

        return "использует Sing";

    }

}

**WakeUpSlap.java**

package moves;

import ru.ifmo.se.pokemon.\*;

public class WakeUpSlap extends PhysicalMove {

    public WakeUpSlap() {

        super(Type.FIGHTING, 70, 100);

    }

    @Override

    protected void applyOppDamage(Pokemon def, double damage) {

        Status p\_stat = def.getCondition();

        if (p\_stat.equals(Status.SLEEP)) {

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

            def.setCondition(e);

            super.applyOppDamage(def, damage\*2);

        }

        else {

            super.applyOppDamage(def, damage);

        }

    }

    @Override

    protected String describe() {

        return "использует Wake-Up Slap";

    }

}

**Thunder.java**

package moves;

import ru.ifmo.se.pokemon.\*;

public class Thunder extends SpecialMove {

    public Thunder() {

        super(Type.ELECTRIC, 110, 70);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        if (Math.random() < 0.3) {

            Effect.paralyze(p);

        }

    }

    @Override

    protected String describe() {

        return "использует Thunder";

    }

}

Результат работы:

Seviper Батя из команды красных вступает в бой!

Golisopod Дед из команды полосатых вступает в бой!

Seviper Батя промахивается

Golisopod Дед промахивается

Seviper Батя промахивается

Golisopod Дед промахивается

Seviper Батя использует Swagger.

Golisopod Дед увеличивает атаку.

Golisopod Дед использует Liquidation.

Seviper Батя теряет 8 здоровья.

Seviper Батя промахивается

Golisopod Дед использует Facade.

Seviper Батя теряет 5 здоровья.

Seviper Батя теряет сознание.

Wimpod Брат из команды красных вступает в бой!

Wimpod Брат использует Facade.

Golisopod Дед теряет 5 здоровья.

Golisopod Дед промахивается

Wimpod Брат использует Facade.

Golisopod Дед теряет 4 здоровья.

Golisopod Дед использует Waterfall.

Wimpod Брат теряет 3 здоровья.

Golisopod Дед промахивается

Golisopod Дед промахивается

Wimpod Брат промахивается

Golisopod Дед растерянно попадает по себе.

Golisopod Дед теряет 3 здоровья.

Wimpod Брат промахивается

Golisopod Дед промахивается

Wimpod Брат использует Waterfall.

Golisopod Дед теряет 3 здоровья.

Golisopod Дед теряет сознание.

Cleffa Дядя из команды полосатых вступает в бой!

Wimpod Брат промахивается

Cleffa Дядя использует Sing.

Wimpod Брат засыпает

Wimpod Брат промахивается

Cleffa Дядя использует Sing.

Wimpod Брат засыпает

Wimpod Брат промахивается

Cleffa Дядя использует Sing.

Wimpod Брат засыпает

Cleffa Дядя использует Flamethrower.

Wimpod Брат теряет 5 здоровья.

Cleffa Дядя использует Flamethrower.

Wimpod Брат теряет 7 здоровья.

Wimpod Брат теряет сознание.

Clefairy Зять из команды красных вступает в бой!

Clefairy Зять использует Wake-Up Slap.

Cleffa Дядя теряет 3 здоровья.

Cleffa Дядя использует Sing.

Clefairy Зять засыпает

Clefairy Зять использует Flamethrower.

Cleffa Дядя теряет 4 здоровья.

Cleffa Дядя использует Flamethrower.

Clefairy Зять теряет 8 здоровья.

Clefairy Зять использует Flamethrower.

Критический удар!

Cleffa Дядя теряет 9 здоровья.

Cleffa Дядя использует Flamethrower.

Clefairy Зять теряет 5 здоровья.

Clefairy Зять использует Flamethrower.

Cleffa Дядя теряет 4 здоровья.

Cleffa Дядя теряет сознание.

Clefable Сосед из команды полосатых вступает в бой!

Clefairy Зять использует Sing.

Clefable Сосед засыпает

Clefable Сосед использует Sing.

Clefairy Зять засыпает

Clefable Сосед использует Sing.

Clefable Сосед использует Flamethrower.

Clefairy Зять теряет 4 здоровья.

Clefairy Зять теряет сознание.

В команде красных не осталось покемонов.

Команда полосатых побеждает в этом бою!

Вывод:

Во время выполнения данной лабораторной работы я изучил основные принципы ООП, научился работать с методами, классами, модификаторами доступами и сторонними библиотеками.