

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное автономное
образовательное учреждение национальный
исследовательский университет «ИТМО»
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА № 2
по дисциплине
‘ПРОГРАММИРОВАНИЕ’

Вариант №78642

Выполнил:
Студент группы Р3112
Кобелев Роман
Павлович
Преподаватель:
Гаврилов Антон
Валерьевич

ИТМО

Санкт-Петербург, 2022

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](#) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](#).

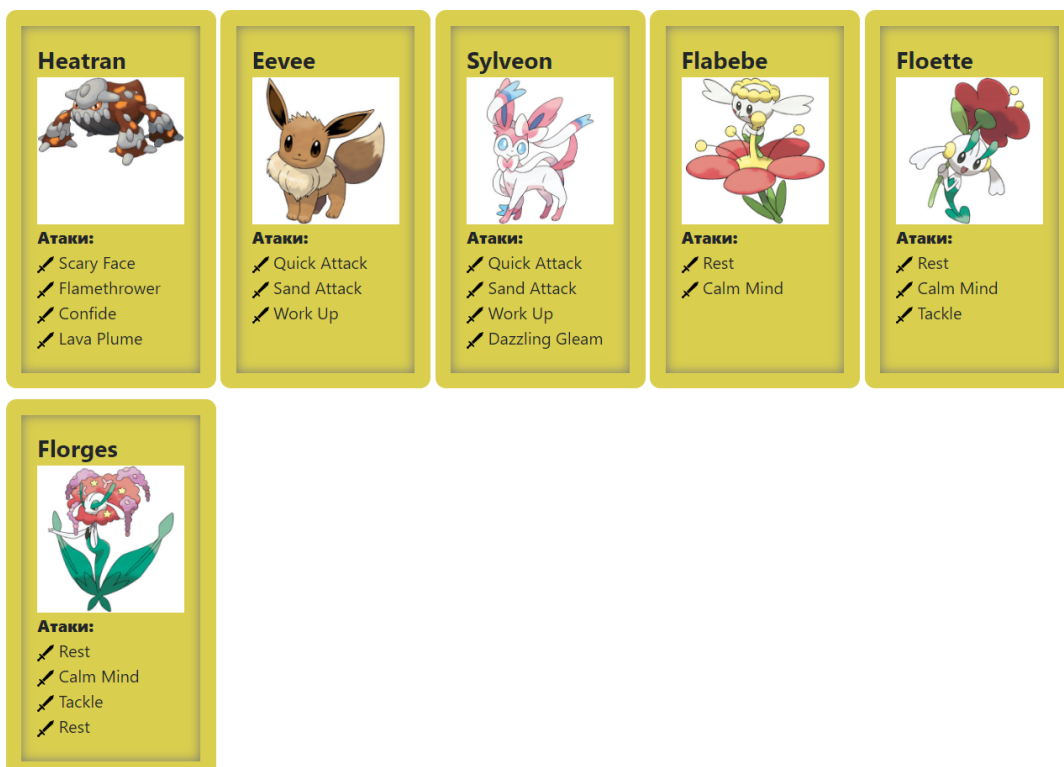
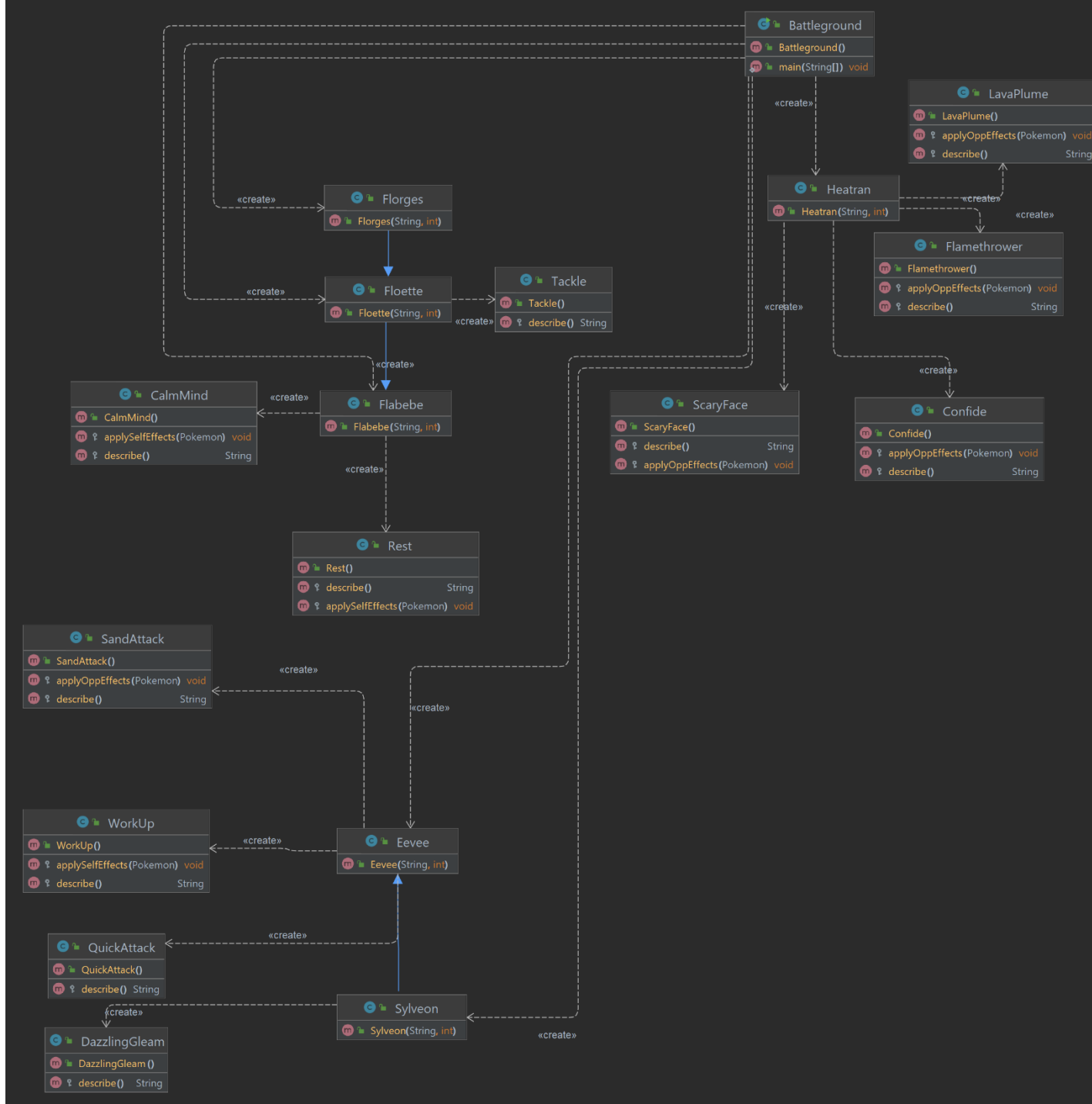


Диаграмма классов реализованной объектной модели:



Исходный код программы

Battleground.java:

```
package added;

import added.pokemon.Eevee;
import ru.ifmo.se.pokemon.*;
import added.pokemon.*;

public class Battleground {
    public static void main(String[] args) {
        Battle b = new Battle();
        b.addAlly(new Heatran("", 1));
        b.addAlly(new Eevee("", 1));
        b.addAlly(new Sylveon("", 1));
        b.addFoe(new Flabebe("", 1));
        b.addFoe(new Floette("", 1));
        b.addFoe(new Florges("", 1));
        b.go();
    }
}
```

Added.moves

CalmMind.java:

```
package added.moves;
import ru.ifmo.se.pokemon.*;
public class CalmMind extends StatusMove{
    public CalmMind() {
        super(Type.PSYCHIC, 0, 100);
    }
    @Override
    protected void applySelfEffects(Pokemon p) {
        Effect e = new Effect().stat(Stat.SPECIAL_ATTACK, +1).stat(Stat.SPECIAL_DEFENSE,
+1);
        p.addEffect(e);
    }
    @Override
    protected String describe() {
        return "uses Calm Mind";
    }
}
```

Confide.java:

```
package added.moves;
import ru.ifmo.se.pokemon.*;
public class Confide extends StatusMove{
    public Confide() {
        super(Type.NORMAL, 0, 100);
    }
    @Override
    protected void applyOppEffects(Pokemon p) {
        Effect e = new Effect().stat(Stat.SPECIAL_ATTACK, -1);
        p.addEffect(e);
    }
    @Override
    protected String describe() {
        return "uses Confide";
    }
}
```

```
}
```

DazzlingGleam.java:

```
package added.moves;
import ru.ifmo.se.pokemon.*;

public class DazzlingGleam extends SpecialMove{
    public DazzlingGleam() {
        super(Type.FAIRY, 80, 1);
    }

    @Override
    protected String describe() {
        return "uses Dazzling Gleam";
    }
}
```

Flamethrower.java:

```
package added.moves;
import ru.ifmo.se.pokemon.*;
public class Flamethrower extends SpecialMove{
    public Flamethrower() {
        super(Type.FIRE, 90, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        Effect e = new Effect().condition(Status.BURN).chance(0.1);
        p.addEffect(e);
    }

    @Override
    protected String describe() {
        return "uses Flamethrower";
    }
}
```

LavaPlume.java:

```
package added.moves;
import ru.ifmo.se.pokemon.*;
public class LavaPlume extends SpecialMove{
    public LavaPlume() {
        super(Type.FIRE, 80, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        Effect e = new Effect().condition(Status.BURN).chance(0.3);
        p.addEffect(e);
    }

    @Override
    protected String describe() {
        return "uses Lava Plume";
    }
}
```

QuickAttack.java:

```
package added.moves;
import ru.ifmo.se.pokemon.*;
public class QuickAttack extends PhysicalMove{
    public QuickAttack() {
        super(Type.NORMAL, 40, 100, 1, 10);
    }
    @Override
    protected String describe() {
        return "uses Quick Attack";
    }
}
```

Rest.java:

```
package added.moves;
import ru.ifmo.se.pokemon.*;
public class Rest extends StatusMove{
    public Rest() {
        super(Type.PSYCHIC, 0, 100);
    }
    @Override
    protected void applySelfEffects(Pokemon p) {
        Effect e = new Effect().condition(Status.SLEEP).turns(2);
        p.addEffect(e);
        Effect e1 = new Effect().stat(Stat.HP, (int) (p.getStat(Stat.HP) -
p.getHP()).turns(2);
        p.addEffect(e1);
    }
    @Override
    protected String describe() {
        return "uses Rest";
    }
}
```

SandAttack.java:

```
package added.moves;
import ru.ifmo.se.pokemon.*;
public class SandAttack extends StatusMove{
    public SandAttack() {
        super(Type.GROUND, 0, 100);
    }
    @Override
    protected void applyOppEffects(Pokemon p) {
        if (!p.hasType(Type.FLYING)) {
            Effect e = new Effect().stat(Stat.ACCURACY, -1);
            p.addEffect(e);
        }
    }
    @Override
    protected String describe() {
        return "uses Sand Attack";
    }
}
```

ScaryFace.java:

```
package added.moves;
import ru.ifmo.se.pokemon.*;
public class ScaryFace extends StatusMove{
    public ScaryFace() {
        super(Type.NORMAL, 0, 100);
    }
    @Override
    protected void applyOppEffects(Pokemon p){
        Effect e = new Effect().stat(Stat.SPEED, -2);
        p.addEffect(e);
    }
    @Override
    protected String describe(){
        return "uses Scary Face";
    }
}
```

Tackle.java:

```
package added.moves;
import ru.ifmo.se.pokemon.*;
public class Tackle extends PhysicalMove{
    public Tackle() {
        super(Type.NORMAL, 40, 100);
    }
    @Override
    protected String describe(){
        return "uses Tackle";
    }
}
```

WorkUp.java:

```
package added.moves;
import ru.ifmo.se.pokemon.*;
public class WorkUp extends StatusMove{
    public WorkUp() {
        super(Type.NORMAL, 0, 100);
    }
    @Override
    protected void applySelfEffects(Pokemon p){
        Effect e = new Effect().stat(Stat.SPECIAL_ATTACK, +1).stat(Stat.SPECIAL_DEFENSE,
+1);
        p.addEffect(e);
    }
    @Override
    protected String describe(){
        return "uses Work Up";
    }
}
```

Added.pokemon

Eevee.java:

```
package added.pokemon;

import ru.ifmo.se.pokemon.*;
import added.moves.QuickAttack;
import added.moves.SandAttack;
import added.moves.WorkUp;
public class Eevee extends Pokemon {
    public Eevee(String name, int level){
        super(name, level);
        setStats(55, 55, 50, 45, 65, 55);
        setType(Type.NORMAL);
        setMove(new QuickAttack(), new SandAttack(), new WorkUp());
    }
}
```

Flabebe.java:

```
package added.pokemon;

import ru.ifmo.se.pokemon.*;
import added.moves.Rest;
import added.moves.CalmMind;
public class Flabebe extends Pokemon{
    public Flabebe(String name, int level){
        super(name, level);
        setStats(44, 38, 39, 61, 79, 42);
        setType(Type.FAIRY);
        setMove(new Rest(), new CalmMind());
    }
}
```

Floette.java

```
package added.pokemon;

import added.moves.Tackle;
import ru.ifmo.se.pokemon.Type;

public class Floette extends Flabebe{
    public Floette (String name, int level){
        super(name, level);
        setStats(54, 45, 47, 75, 98, 52);
        setType(Type.FAIRY);
        addMove(new Tackle());
    }
}
```

Florges.java

```
package added.pokemon;

import ru.ifmo.se.pokemon.Type;

public class Florges extends Floette{
    public Florges(String name, int level){
        super(name, level);
        setStats(78, 65, 68, 112, 154, 75);
        addType(Type.FAIRY);
    }
}
```


Heatren.java

```
package added.pokemon;

import ru.ifmo.se.pokemon.*;
import added.moves.LavaPlume;
import added.moves.Confide;
import added.moves.Flamethrower;
import added.moves.ScaryFace;

public class Heatran extends Pokemon{
    public Heatran (String name, int level){
        super(name, level);
        setStats(91, 90, 106, 130, 106, 77);
        setType(Type.FIRE, Type.STEEL);
        setMove(new LavaPlume(), new Confide(), new Flamethrower(), new ScaryFace());
    }
}
```

Sylveon.java

```
package added.pokemon;

import added.moves.DazzlingGleam;
import ru.ifmo.se.pokemon.Type;

public class Sylveon extends Eevee{
    public Sylveon(String name, int level){
        super(name, level);
        setStats(95, 65, 65, 110, 130, 60);
        setType(Type.FAIRY);
        addMove(new DazzlingGleam());
    }
}
```

Результат работы:

Результат 1.

Heatran из команды желтых вступает в бой!

Flabebe из команды зеленых вступает в бой!

Heatran uses Lava Plume.

Flabebe теряет 9 здоровья.

Flabebe uses Calm Mind.

Heatran uses Lava Plume.

Flabebe теряет 6 здоровья.

Flabebe теряет сознание.

Floette из команды зеленых вступает в бой!

Heatran uses Confide.

Floette uses Rest.

Floette засыпает

Heatran uses Flamethrower.

Floette теряет 8 здоровья.

Floette uses Rest.

Floette теряет 8 здоровья.

Florges из команды зеленых вступает в бой!

Heatran uses Lava Plume.

Florges теряет 8 здоровья.

Florges uses Rest.

Florges засыпает

Florges теряет 8 здоровья.

В команде зеленых не осталось покемонов.

Команда желтых побеждает в этом бою!

Результат 2.

Heatran из команды желтых вступает в бой!

Flabebe из команды синих вступает в бой!

Heatran uses Lava Plume.

Flabebe теряет 7 здоровья.

Flabebe uses Calm Mind.

Heatran uses Confide.

Flabebe uses Calm Mind.

Heatran uses Flamethrower.

Flabebe теряет 5 здоровья.

Flabebe uses Calm Mind.

Heatran uses Lava Plume.

Flabebe теряет 8 здоровья.

Flabebe теряет сознание.

Floette из команды синих вступает в бой!

Heatran uses Scary Face.

Floette uses Tackle.

Критический удар!

Heatran теряет 3 здоровья.

Floette uses Calm Mind.

Heatran uses Flamethrower.

Floette теряет 7 здоровья.

Floette uses Calm Mind.

Heatran uses Lava Plume.

Floette теряет 8 здоровья.

Floette теряет сознание.

Florges из команды синих вступает в бой!

Heatran uses Confide.

Florges uses Calm Mind.

Heatran uses Flamethrower.

Florges теряет 8 здоровья.

Florges uses Tackle.

Heatran теряет 2 здоровья.

Heatran uses Flamethrower.

Florges теряет 7 здоровья.

Florges теряет сознание.

В команде синих не осталось покемонов.

Команда желтых побеждает в этом бою!

Вывод:

В ходе выполнения данной лабораторной работы я познакомился с основными концепциями ООП. Научился подключать внешние jar-архивы, необходимые для компиляции и выполнения программы.