

Санкт-Петербургский Национальный Исследовательский
Университет ИТМО
Мегафакультет Компьютерных Технологий и Управления
Факультет Программной Инженерии и Компьютерной Техники



Вариант №666666
Лабораторная работа №2
по дисциплине
Программирование

Выполнил Студент группы number
Student's name
Преподаватель:
Teacher's name

г. Санкт-Петербург
2021г.

Содержание

1	Текст задания	3
2	Покемоны	3
3	Диаграмма классов	4
4	Исходный код программы	4
5	Результат выполнения:	9
6	Вывод	10

1 Текст задания

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.

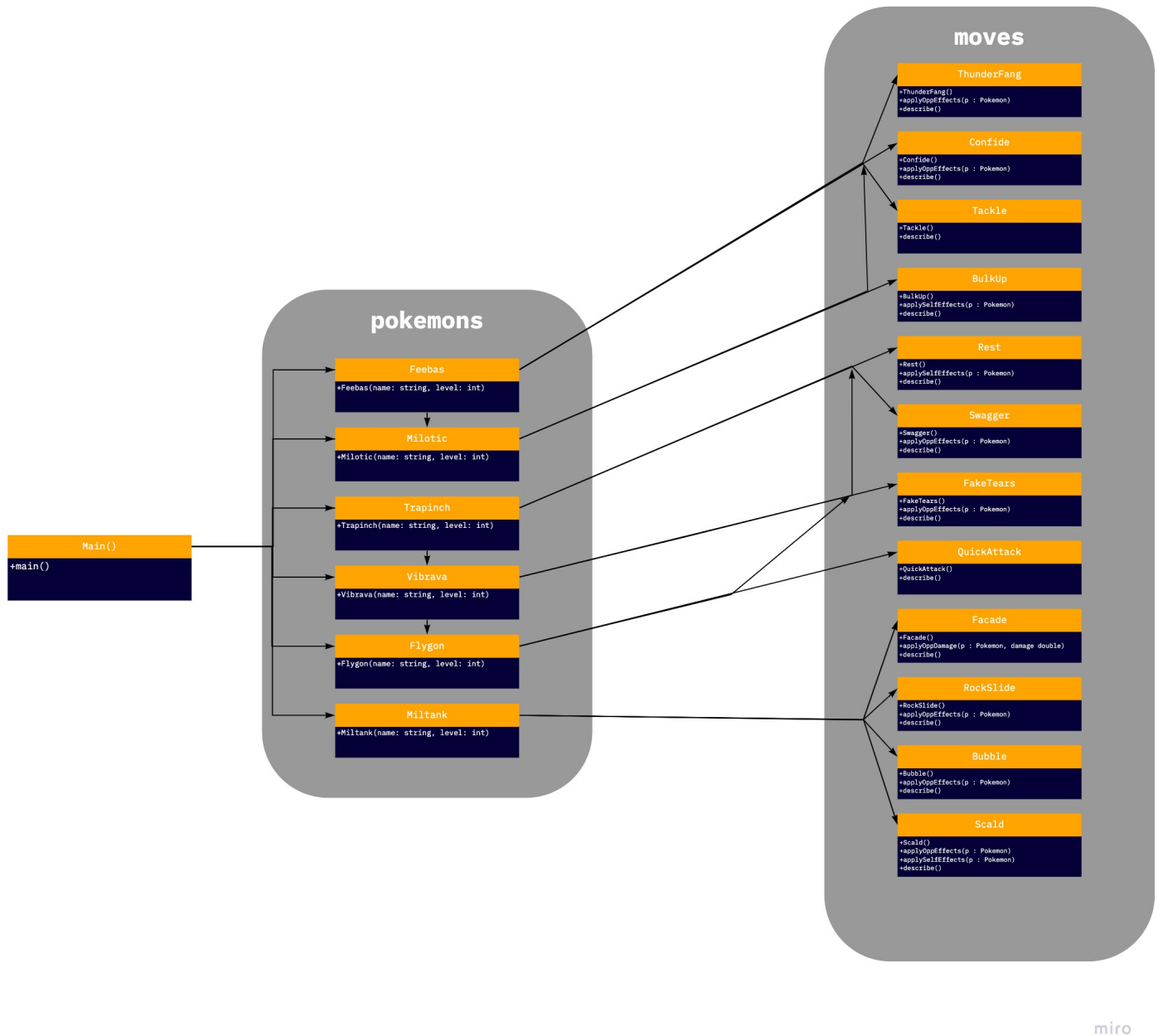
Базовые классы и симулятор сражения находятся в jar-архиве (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](#).

Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах <http://pokemon-universe.ru>, <http://pokemondb.net>, <http://veekun.com/dex/pokemon>

2 Покемоны



3 Диаграмма классов



4 Исходный код программы

Main.java

```
1 import pokemons.*;
2 import ru.ifmo.se.pokemon.Battle;
3 import ru.ifmo.se.pokemon.Pokemon;
4
5 public class Main {
6     public static void main(String[] args) {
7         Battle b = new Battle();
8         b.addAlly(new Feebas("Дирихле",1));
9         b.addAlly(new Flygon("Коши",1));
10        b.addAlly(new Milotic("Лагранж",1));
11        b.addFoe(new Miltank("Вейерштрасс",1));
12        b.addFoe(new Trapinch("Кантор",1));
13        b.addFoe(new Vibrava("Гайсс",1));
```

```

14         b.go();
15     }
16 }

```

Pokemons

Feebas.java

```

1  package pokemons;
2
3  import moves.*;
4  import ru.ifmo.se.pokemon.Pokemon;
5  import ru.ifmo.se.pokemon.Type;
6
7  public class Feebas extends Pokemon {
8      public Feebas(String name, int level){
9          super(name,level);
10         setStats(20,15,20, 10, 55,80);
11         setType(Type.WATER);
12         setMove(new ThunderFang(), new Confide(), new Tackle());
13     }
14 }

```

Flygon.java

```

1  package pokemons;
2
3  import moves.QuickAttack;
4  import ru.ifmo.se.pokemon.Type;
5
6  public class Flygon extends Vibrava{
7      public Flygon(String name, int level){
8          super(name,level);
9          setStats(80,100,80, 80, 80,100);
10         setType(Type.GROUND, Type.DRAGON);
11         addMove(new QuickAttack());
12     }
13 }

```

Milotic.java

```

1  package pokemons;
2
3  import moves.BulkUp;
4  import moves.Confide;
5  import moves.Tackle;
6  import moves.ThunderFang;
7  import ru.ifmo.se.pokemon.Type;
8
9  public class Milotic extends Feebas {
10     public Milotic(String name, int level){
11         super(name,level);
12         setStats(95,60,79, 100, 125,81);
13         addMove(new BulkUp());
14     }
15 }

```

Miltank.java

```

1  package pokemons;
2
3  import moves.Bubble;
4  import moves.Facade;
5  import moves.RockSlide;
6  import moves.Scald;
7  import ru.ifmo.se.pokemon.Pokemon;
8  import ru.ifmo.se.pokemon.Type;
9
10 public class Miltank extends Pokemon {
11     public Miltank(String name, int level){
12         super(name,level);
13         setStats(95,80,105, 40, 70,100);
14         setType(Type.NORMAL);
15         setMove(new Bubble(), new Facade(), new Scald(), new RockSlide());
16     }
17 }

```

Trapinch.java

```

1 package pokemons;
2
3 import moves.*;
4 import ru.ifmo.se.pokemon.Pokemon;
5 import ru.ifmo.se.pokemon.Type;
6
7 public class Trapinch extends Pokemon {
8     public Trapinch(String name, int level){
9         super(name,level);
10        setStats(45,100,45, 45, 45,10);
11        setType(Type.GROUND);
12        setMove(new Rest(),new Swagger());
13    }
14 }

```

Vibrava.java

```

1 package pokemons;
2
3 import moves.FakeTears;
4 import ru.ifmo.se.pokemon.Type;
5
6 public class Vibrava extends Trapinch{
7     public Vibrava(String name, int level){
8         super(name,level);
9         setStats(50,70,50, 50, 50,70);
10        setType(Type.GROUND, Type.DRAGON);
11        addMove(new FakeTears());
12    }
13 }

```

Moves

Bubble.java

```

1 package moves;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class Bubble extends SpecialMove {
6     public Bubble(){
7         super(Type.WATER,40,100);
8     }
9
10    @Override
11    protected void applyOppEffects(Pokemon p) {
12        p.addEffect(new Effect().chance(0.1).stat(Stat.SPEED,-1));
13    }
14
15    @Override
16    protected String describe() {
17        return "using Bubble";
18    }
19 }

```

BulkUp.java

```

1 package moves;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class BulkUp extends StatusMove {
6     public BulkUp(){
7         super(Type.FIGHTING, 0 ,0);
8     }
9
10    @Override
11    public void applySelfEffects(Pokemon p) {
12        p.addEffect(new Effect().turns(0).stat(Stat.ATTACK, +1).stat(Stat.DEFENSE,+1));
13    }
14
15    @Override
16    protected String describe() {
17        return "using BulkUp";
18    }
19 }

```

Confide.java

```

1 package moves;

```

```

2
3 import ru.ifmo.se.pokemon.*;
4
5 public class Confide extends StatusMove {
6     public Confide() {
7         super(Type.NORMAL, 0, 0);
8     }
9
10    @Override
11    protected void applyOppEffects(Pokemon p) {
12        p.addEffect(new Effect().turns(0).stat(Stat.SPECIAL_ATTACK, -1));
13    }
14
15    @Override
16    protected String describe() {
17        return "using Confide";
18    }
19 }

```

Facade.java

```

1 package moves;
2
3 import ru.ifmo.se.pokemon.PhysicalMove;
4 import ru.ifmo.se.pokemon.Pokemon;
5 import ru.ifmo.se.pokemon.Status;
6 import ru.ifmo.se.pokemon.Type;
7
8 public class Facade extends PhysicalMove {
9     public Facade(){
10        super(Type.NORMAL, 70, 100);
11    }
12    @Override
13    protected void applyOppDamage(Pokemon p, double damage) {
14        Status p_stat = p.getCondition();
15        if (p_stat.equals(Status.BURN) || p_stat.equals(Status.POISON) || p_stat.equals(
16            Status.PARALYZE)) {
17            super.applyOppDamage(p, damage*2);
18        }
19        else {
20            super.applyOppDamage(p, damage);
21        }
22    }
23    @Override
24    protected String describe() {
25        return "using Facade";
26    }
27 }

```

FakeTears.java

```

1 package moves;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class FakeTears extends StatusMove {
6     public FakeTears(){
7         super(Type.DARK, 0, 100);
8     }
9     @Override
10    protected void applyOppEffects(Pokemon p) {
11        p.addEffect(new Effect().turns(0).stat(Stat.SPECIAL_DEFENSE, -2));
12    }
13    @Override
14    protected String describe() {
15        return "using Fake Tears";
16    }
17 }

```

QuickAttack.java

```

1 package moves;
2
3 import ru.ifmo.se.pokemon.PhysicalMove;
4 import ru.ifmo.se.pokemon.Type;
5
6 public class QuickAttack extends PhysicalMove {
7     public QuickAttack(){

```

```

8         super(Type.NORMAL,40,100,1,1);
9     }
10    @Override
11    protected String describe() {
12        return "using Quick Attack";
13    }
14 }

```

Rest.java

```

1 package moves;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class Rest extends StatusMove {
6     public Rest(){
7         super(Type.PSYCHIC,0,0);
8     }
9
10    protected void applySelfEffects(Pokemon p){
11        p.addEffect(new Effect().turns(2).condition(Status.SLEEP));
12        p.setCondition(new Effect().turns(0).stat(Stat.HP, -99999));
13    }
14
15    @Override
16    protected String describe() {
17        return "решил отдохнуть";
18    }
19 }

```

RockSlide.java

```

1 package moves;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class RockSlide extends PhysicalMove {
6     public RockSlide(){
7         super(Type.ROCK,75,90);
8     }
9     @Override
10    public void applyOppEffects(Pokemon p){
11        if (Math.random()<=0.3) {
12            Effect.flinch(p);
13        }
14    }
15    @Override
16    protected String describe() {
17        return "using Rock Slide";
18    }
19 }

```

Scald.java

```

1 package moves;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class Scald extends SpecialMove {
6     public Scald(){
7         super(Type.WATER,80,100);
8     }
9
10    @Override
11    protected void applyOppEffects(Pokemon p) {
12        if (Math.random() <= 0.3){
13            Effect.burn(p);
14        }
15    }
16    protected void applySelfEffects(Pokemon p){
17        Status self_st = p.getCondition();
18        if (self_st.equals(Status.FREEZE)){
19            p.addEffect(new Effect().condition(Status.NORMAL));
20        }
21    }
22
23    @Override
24    protected String describe() {

```



```

25         return "using_Scald";
26     }
27 }

```

Swagger.java

```

1 package moves;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class Swagger extends StatusMove {
6     public Swagger(){
7         super(Type.NORMAL, 0,85);
8     }
9
10    @Override
11    protected void applyOppEffects(Pokemon p){
12        p.confuse();
13        p.setMod(Stat.ATTACK, 2);
14    }
15
16    @Override
17    protected String describe() {
18        return "using_Swagger";
19    }
20
21 }

```

Tackle.java

```

1 package moves;
2
3 import ru.ifmo.se.pokemon.PhysicalMove;
4 import ru.ifmo.se.pokemon.Type;
5
6 public class Tackle extends PhysicalMove {
7     public Tackle(){
8         super(Type.NORMAL,40,100);
9     }
10
11    @Override
12    protected String describe() {
13        return "using_Tackle";
14    }
15 }

```

ThunderFang.java

```

1 package moves;
2
3 import ru.ifmo.se.pokemon.Effect;
4 import ru.ifmo.se.pokemon.PhysicalMove;
5 import ru.ifmo.se.pokemon.Pokemon;
6 import ru.ifmo.se.pokemon.Type;
7
8 public class ThunderFang extends PhysicalMove {
9     public ThunderFang(){
10         super(Type.ELECTRIC,65,95);
11     }
12
13    @Override
14    public void applyOppEffects(Pokemon p){
15        if (Math.random() <=0.1) {
16            Effect.flinch(p);
17        }
18        if (Math.random() <=0.1){
19            Effect.paralyze(p);
20        }
21    }
22
23    @Override
24    protected String describe() {
25        return "using_Thunder_Fang";
26    }
27 }

```

5 Результат выполнения:

Feebas Дирихле из команды фиолетовых вступает в бой!

Miltank Вейерштрасс из команды зеленых вступает в бой!
Miltank Вейерштрасс using Scald.
Feebas Дирихле теряет 2 здоровья.
Feebas Дирихле воспламеняется

Feebas Дирихле using Thunder Fang.
Miltank Вейерштрасс восстанавливает 1 здоровья.

Feebas Дирихле using Tackle.
Miltank Вейерштрасс восстанавливает 1 здоровья.

Miltank Вейерштрасс using Bubble.
Feebas Дирихле теряет 1 здоровья.

Feebas Дирихле using Tackle.
Miltank Вейерштрасс восстанавливает 1 здоровья.

Miltank Вейерштрасс using Rock Slide.
Feebas Дирихле теряет 4 здоровья.

Feebas Дирихле using Thunder Fang.
Miltank Вейерштрасс восстанавливает 1 здоровья.

Miltank Вейерштрасс using Scald.
Feebas Дирихле теряет 3 здоровья.

Feebas Дирихле using Thunder Fang.
Miltank Вейерштрасс восстанавливает 1 здоровья.
Miltank Вейерштрасс парализован

Miltank Вейерштрасс using Rock Slide.
Критический удар!
Feebas Дирихле теряет 11 здоровья.
Feebas Дирихле теряет сознание.
Flygon Коши из команды фиолетовых вступает в бой!
Miltank Вейерштрасс using Facade.
Критический удар!
Flygon Коши теряет 11 здоровья.

Flygon Коши промахивается

Flygon Коши using Quick Attack.
Miltank Вейерштрасс теряет 3 здоровья.

Miltank Вейерштрасс using Scald.
Критический удар!
Flygon Коши теряет 13 здоровья.
Flygon Коши воспламеняется
Flygon Коши теряет сознание.
Milotic Лагранж из команды фиолетовых вступает в бой!
Miltank Вейерштрасс using Rock Slide.
Критический удар!
Milotic Лагранж теряет 9 здоровья.

Milotic Лагранж using Thunder Fang.
Miltank Вейерштрасс теряет 4 здоровья.

Miltank Вейерштрасс using Facade.
Критический удар!
Milotic Лагранж теряет 12 здоровья.
Milotic Лагранж теряет сознание.
В команде фиолетовых не осталось покемонов.
Команда зеленых побеждает в этом бою!

6 Вывод

В процессе выполнения этой лабораторной работы я познакомился с основными принципами объектно-ориентированное программирование в языке Java. Научился работать с методами, классами, модификаторами доступа и сторонними библиотеками.