

Санкт-Петербургский Национальный Исследовательский
Университет ИТМО
Факультет Программной Инженерии и Компьютерной Техники



УНИВЕРСИТЕТ ИТМО

Вариант №311817
Лабораторная работа №2
по дисциплине
"Программирование"

Выполнила студентка группы Р3117
Галина Игнатова
Преподаватель:
Письмак Алексей Евгеньевич

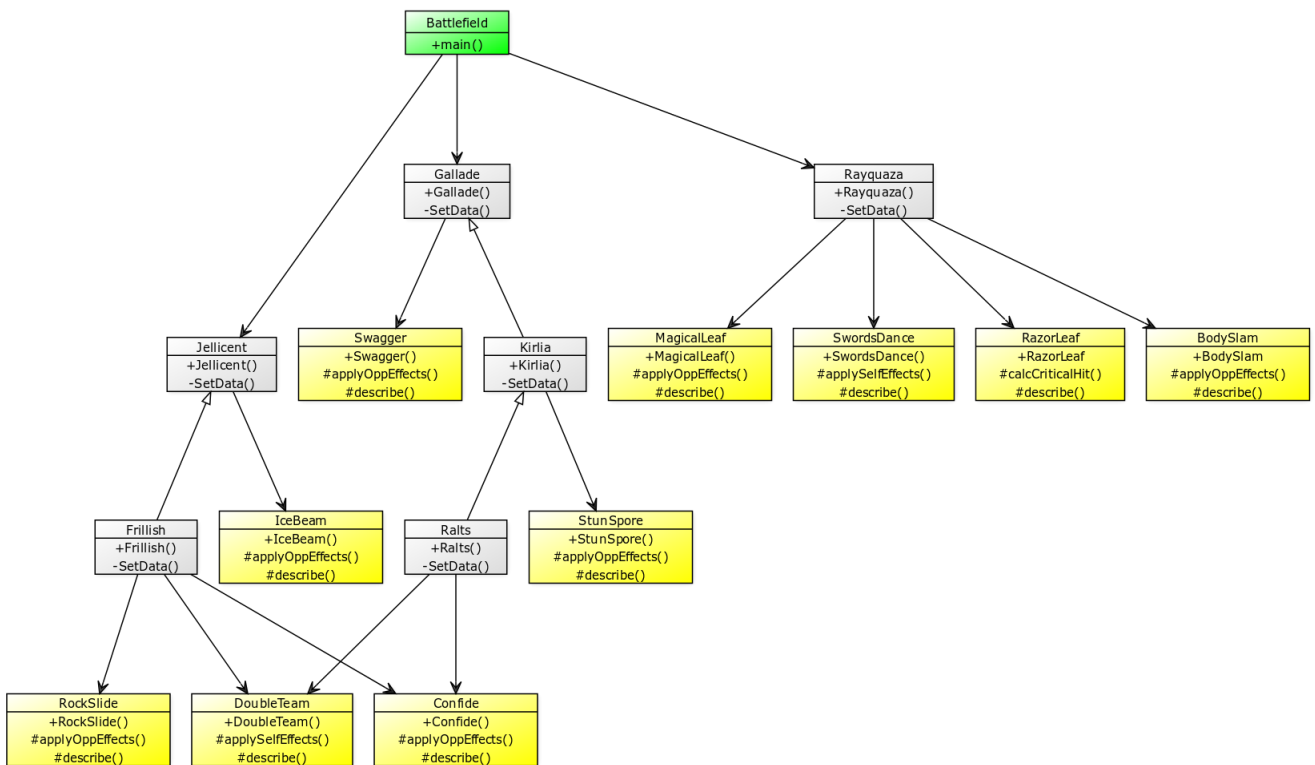
г. Санкт-Петербург
2021г.

1 Текст задания

Ваши покемоны:



2 Диаграмма классов реализованной объектной модели



CREATED WITH YUML

3 Исходный код программы

Листинг 1: Battlefield.java

```
1 import Pokemons.*;
2 import ru.ifmo.se.pokemon.*;
3
4 public class Battlefield {
5     public static void main(String[] args) {
```

```

6      Battle b = new Battle();
7      Frillish frillish = new Frillish("Suga", 1);
8      Gallade gallade = new Gallade("Jimin", 1);
9      Jellicent jellicent = new Jellicent("Jungkook", 1);
10     Kirlia kirlia = new Kirlia("Taehyung", 2);
11     Ralts ralts = new Ralts("Hoseok", 1);
12     Rayquaza rayquaza = new Rayquaza("Namjoon", 1);
13     b.addAlly(frillish);
14     b.addAlly(gallade);
15     b.addAlly(jellicent);
16     b.addFoe(kirlia);
17     b.addFoe(ralts);
18     b.addFoe(rayquaza);
19     b.go();
20 }
21 }

```

Листинг 2: Pokemons/Frillish.java

```

1  package Pokemons;
2
3  import Moves.Physical.*;
4  import Moves.Status.*;
5  import ru.ifmo.se.pokemon.*;
6
7  public class Frillish extends Pokemon {
8      public Frillish(String name, int level) {
9          super(name, level);
10         setData();
11     }
12
13     private void setData() {
14         setStats(55, 40, 50, 65, 85, 40);
15         setType(Type.WATER, Type.GHOST);
16         setMove(new Confide(), new RockSlide(), new DoubleTeam());
17     }
18 }

```

Листинг 3: Pokemons/Gallade.java

```

1  package Pokemons;
2
3  import Moves.Status.*;
4  import ru.ifmo.se.pokemon.*;
5
6  public class Gallade extends Kirlia {
7      public Gallade(String name, int level) {
8          super(name, level);
9          setData();
10     }
11
12     private void setData() {
13         setStats(68, 125, 65, 65, 115, 80);

```

```

14     setType(Type.PSYCHIC, Type.FIGHTING);
15     setMove(new Swagger());
16 }
17 }

```

Листинг 4: Pokemons/Jellicent.java

```

1 package Pokemons;
2
3 import Moves.Physical.*;
4 import Moves.Special.*;
5 import Moves.Status.*;
6 import ru.ifmo.se.pokemon.*;
7
8 public class Jellicent extends Frillish {
9     public Jellicent(String name, int level) {
10         super(name, level);
11         setData();
12     }
13
14     private void setData() {
15         setStats(100, 60, 70, 85, 105, 60);
16         setType(Type.WATER, Type.GHOST);
17         setMove(new IceBeam());
18     }
19 }

```

Листинг 5: Pokemons/Kirlia.java

```

1 package Pokemons;
2
3 import Moves.Status.*;
4 import ru.ifmo.se.pokemon.*;
5
6 public class Kirlia extends Ralts {
7     public Kirlia(String name, int level) {
8         super(name, level);
9         setData();
10     }
11
12     private void setData() {
13         setStats(38, 35, 35, 65, 55, 50);
14         setType(Type.PSYCHIC, Type.FAIRY);
15         setMove(new StunSpore());
16     }
17 }

```

Листинг 6: Pokemons/Ralts.java

```

1 package Pokemons;
2
3 import Moves.Status.*;
4 import ru.ifmo.se.pokemon.*;
5

```

```

6 public class Ralts extends Pokemon {
7     public Ralts(String name, int level) {
8         super(name, level);
9         setData();
10    }
11
12    private void setData() {
13        setStats(28, 25, 25, 45, 35, 40);
14        setType(Type.PSYCHIC, Type.FAIRY);
15        setMove(new Confide(), new DoubleTeam());
16    }
17 }

```

Листинг 7: Pokemons/Rayquaza.java

```

1 package Pokemons;
2
3 import Moves.Physical.*;
4 import Moves.Special.*;
5 import Moves.Status.*;
6 import ru.ifmo.se.pokemon.*;
7
8 public class Rayquaza extends Pokemon {
9     public Rayquaza(String name, int level) {
10        super(name, level);
11        setData();
12    }
13
14    private void setData() {
15        setStats(105, 150, 90, 150, 90, 95);
16        setType(Type.DRAGON, Type.FLYING);
17        setMove(new BodySlam(), new MagicalLeaf(), new SwordsDance(), new RazorLeaf());
18    }
19 }

```

Листинг 8: Moves/Physical/BodySlam.java

```

1 package Moves.Physical;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class BodySlam extends PhysicalMove {
6     public BodySlam() {
7         super(Type.NORMAL, 85, 1);
8     }
9
10    @Override
11    protected void applyOppEffects(Pokemon opp) {
12        if (Math.random() <= 0.3) Effect.paralyze(opp);
13    }
14
15    @Override
16    protected String describe() {

```

```

17     return "uses_Body_Slam";
18 }
19 }

```

Листинг 9: Moves/Physical/RazorLeaf.java

```

1 package Moves.Physical;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class RazorLeaf extends PhysicalMove {
6     public RazorLeaf(){
7         super(Type.GRASS, 55, 0.95);
8     }
9     @Override
10    protected double calcCriticalHit(Pokemon opp, Pokemon self){
11        return 1 >> 3;
12    }
13    @Override
14    protected String describe(){
15        return "uses_Razor_Leaf";
16    }
17 }

```

Листинг 10: Moves/Physical/RockSlide.java

```

1 package Moves.Physical;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class RockSlide extends PhysicalMove {
6     public RockSlide(){
7         super(Type.ROCK, 75, 0.9);
8     }
9     @Override
10    protected void applyOppEffects(Pokemon opp){
11        if (Math.random() <= 0.3) Effect.flinch(opp);
12    }
13    @Override
14    protected String describe() {
15        return "uses_Rock_Slide";
16    }
17 }

```

Листинг 11: Moves/Special/IceBeam.java

```

1 package Moves.Special;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class IceBeam extends SpecialMove {
6     public IceBeam(){
7         super(Type.ICE, 90, 1);
8     }

```

```

9      @Override
10     protected void applyOppEffects(Pokemon opp){
11         if (Math.random() <= 0.1) Effect.freeze(opp);
12     }
13     @Override
14     protected String describe(){
15         return "uses_Ice_Beam";
16     }
17 }

```

Листинг 12: Moves/Special/MagicalLeaf.java

```

1 package Moves.Special;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class MagicalLeaf extends SpecialMove {
6     public MagicalLeaf() {
7         super(Type.GRASS, 60, Double.MAX_VALUE);
8     }
9
10    @Override
11    protected void applyOppEffects(Pokemon opp) {
12        opp.addEffect(new Effect().stat(Stat.EVASION, 0).stat(Stat.ACCURACY, 1));
13    }
14
15    @Override
16    protected String describe() {
17        return "uses_Magical_Leaf";
18    }
19 }

```

Листинг 13: Moves/Status/Confide.java

```

1 package Moves.Status;
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     @Override
10    protected void applyOppEffects(Pokemon opp){
11        opp.addEffect(new Effect().stat(Stat.SPECIAL_ATTACK , -1));
12    }
13    @Override
14    protected String describe(){
15        return "uses_Confide";
16    }
17 }

```

Листинг 14: Moves/Status/DoubleTeam.java

```

1 package Moves.Status;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class DoubleTeam extends StatusMove {
6     public DoubleTeam(){
7         super(Type.NORMAL, 0, 0);
8     }
9
10    @Override
11    protected void applySelfEffects(Pokemon self){
12        self.addEffect(new Effect().stat(Stat.EVASION , 1));
13    }
14    @Override
15    protected String describe() {
16        return "uses_Double_Team";
17    }
18 }

```

Листинг 15: Moves/Status/StunSpore.java

```

1 package Moves.Status;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class StunSpore extends StatusMove {
6     public StunSpore(){
7         super(Type.GRASS, 0, 75);
8     }
9     @Override
10    protected void applyOppEffects(Pokemon opp){
11        Effect.paralyze(opp);
12    }
13    @Override
14    protected String describe(){
15        return "uses_Stun_Spore";
16    }
17 }

```

Листинг 16: Moves/Status/Swagger.java

```

1 package Moves.Status;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class Swagger extends StatusMove {
6     public Swagger(){super(Type.NORMAL, 0, 0.85);}
7     @Override
8     protected void applyOppEffects(Pokemon opp){
9         Effect.confuse(opp);
10        opp.addEffect(new Effect().stat(Stat.ATTACK , 2));
11    }

```



```

12     @Override
13     protected String describe(){
14         return "uses_Swagger";
15     }
16 }

```

Листинг 17: Moves/Status/SwordsDance.java

```

1 package Moves.Status;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class SwordsDance extends StatusMove {
6     public SwordsDance(){
7         super(Type.NORMAL, 0, 0);
8     }
9     @Override
10    protected void applySelfEffects(Pokemon self){
11        self.addEffect(new Effect().stat(Stat.ATTACK , 2));
12    }
13    @Override
14    protected String describe(){
15        return "uses_Swords_Dance";
16    }
17 }

```

4 Результат выполнения

```

Frillish Suga из команды желтых вступает в бой!
Kirlia Taehyung из команды черных вступает в бой!
Kirlia Taehyung uses Stun Spore.
Frillish Suga парализован
Frillish Suga промахивается
Kirlia Taehyung uses Stun Spore.
Frillish Suga промахивается
Kirlia Taehyung uses Stun Spore.
Frillish Suga промахивается
Kirlia Taehyung uses Stun Spore.
Frillish Suga uses Rock Slide.
Критический удар!
Kirlia Taehyung теряет 8 здоровья.
Kirlia Taehyung uses Stun Spore.
Frillish Suga промахивается
Kirlia Taehyung uses Stun Spore.
Frillish Suga промахивается
Kirlia Taehyung uses Stun Spore.
Kirlia Taehyung uses Stun Spore.
Frillish Suga промахивается
Kirlia Taehyung uses Stun Spore.
Frillish Suga промахивается
Kirlia Taehyung uses Stun Spore.

```

Frillish Suga промахивается
 Kirlia Taehyung uses Stun Spore.
 Frillish Suga промахивается
 Kirlia Taehyung uses Stun Spore.
 Kirlia Taehyung uses Stun Spore.
 Frillish Suga промахивается
 Kirlia Taehyung uses Stun Spore.
 Frillish Suga промахивается
 Kirlia Taehyung uses Stun Spore.
 Frillish Suga промахивается
 Kirlia Taehyung uses Stun Spore.
 Frillish Suga uses Rock Slide.
 Критический удар!
 Kirlia Taehyung теряет 6 здоровья.
 Kirlia Taehyung теряет сознание.
 Ralts Hoseok из команды черных вступает в бой!
 Frillish Suga промахивается
 Ralts Hoseok промахивается
 Frillish Suga uses Rock Slide.
 Критический удар!
 Ralts Hoseok теряет 10 здоровья.
 Ralts Hoseok промахивается
 Frillish Suga uses Rock Slide.
 Критический удар!
 Ralts Hoseok теряет 10 здоровья.
 Ralts Hoseok теряет сознание.
 Rayquaza Namjoon из команды черных вступает в бой!
 Frillish Suga промахивается
 Rayquaza Namjoon uses Razor Leaf.
 Frillish Suga теряет 1 здоровья.
 Frillish Suga uses Rock Slide.
 Критический удар!
 Rayquaza Namjoon теряет 14 здоровья.
 Rayquaza Namjoon теряет сознание.
 В команде черных не осталось покемонов.
 Команда желтых побеждает в этом бою!

5 Вывод

При написании лабораторной работы я изучила основные принципы ООП, научилась подключать внешние jar-файлы к своим программам, делать UML-диаграммы, а также создавать jar-файлы из нескольких классов.