Национальный Исследовательский Университет ИТМО
Факультет Программной Инженерии и Компьютерной Техники

Вариант № 82002
Лабораторная работа № 2
по дисциплине
'Программирование'

Выполнил:
Студент группы P3113
***Куперштейн Дмитрий;* : 269359**
Преподаватель:
ПИСЬМАК АЛЕКСЕЙ ЕВГЕНЬЕВИЧ

Санкт-Петербург 2019 г.

Содержание

1 Задание **3**

2 Диаграмма классов реализованной объектной модели **4**

3 Исходный код программы **5**

3.1 `main.java.com.kupp.progLab2` 5

3.2 `main.java.com.kupp.progLab2.moves` 5

3.3 `main.java.com.kupp.progLab2.pokemons` 10

4 Результат работы программы **12**

5 Вывод **15**

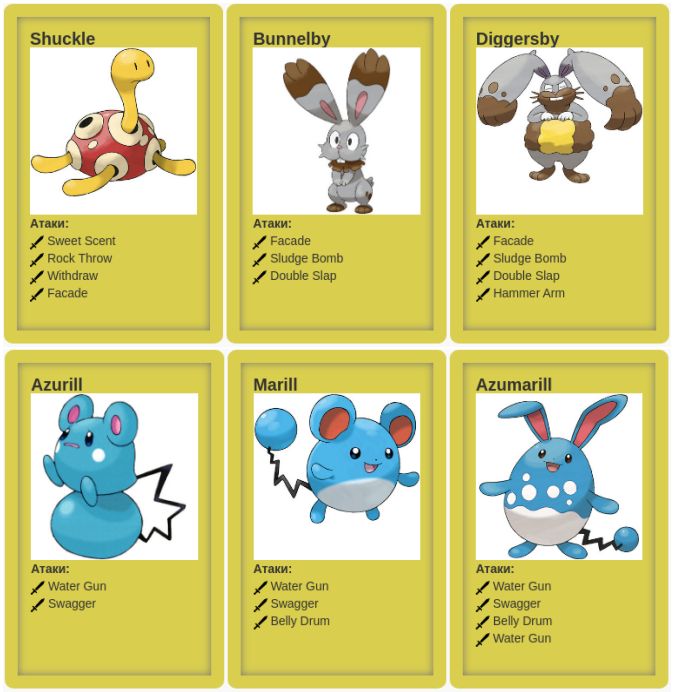
1 Задание

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:

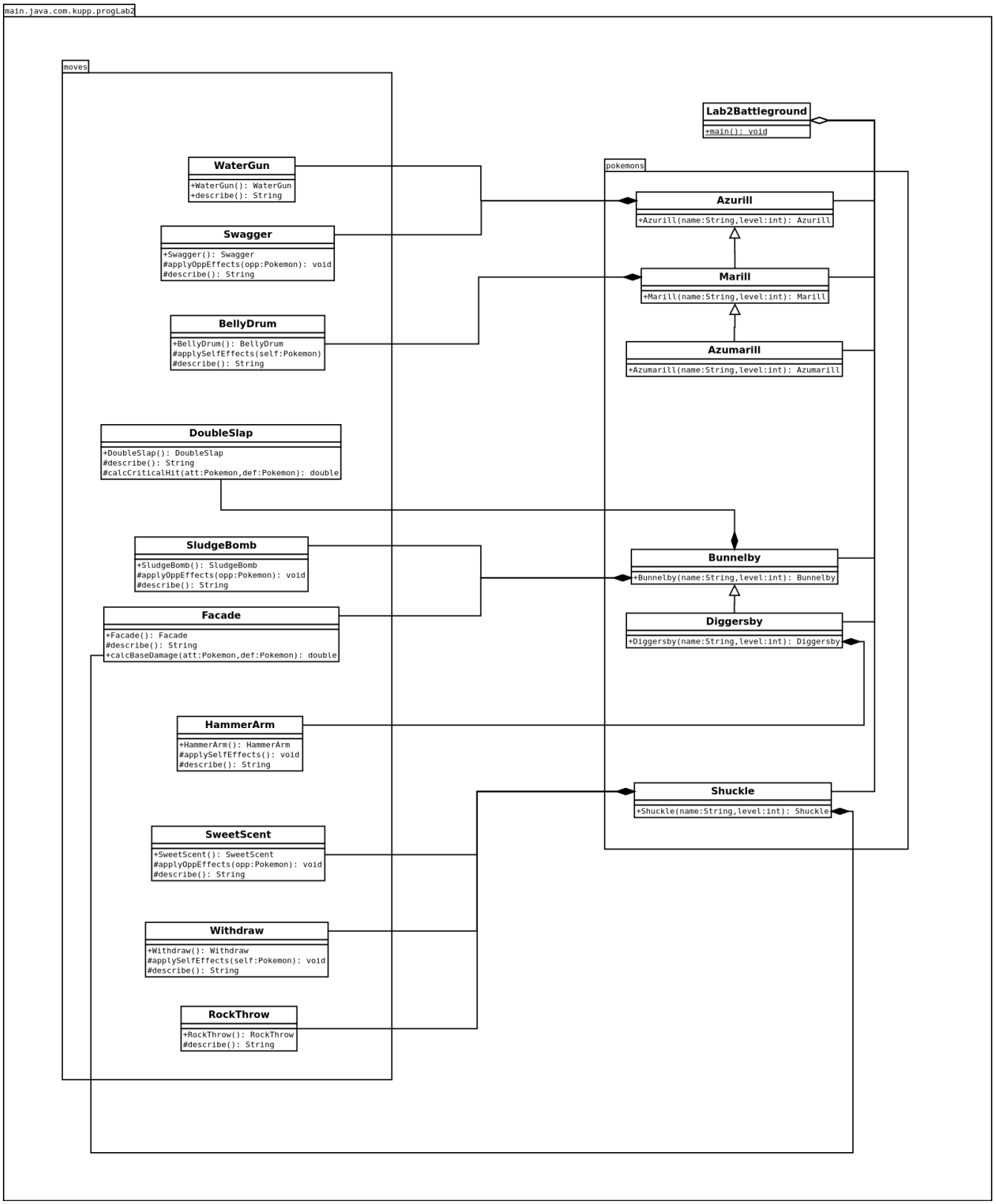
- очки здоровья (HP)
- атака (attack)
- защита (defense)
- специальная атака (special attack)
- специальная защита (special defense)
- скорость (speed)

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.
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 `Pokemon.jar` (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc – `https://se.ifmo.ru/~tony/doc/`.

Покемоны для варианта 82002:



2 Диаграмма классов реализованной объектной модели



3 Исходный код программы

3.1 main.java.com.kupp.progLab2

3.1.1 Lab2Battleground.java

```java
package main.java.com.kupp.progLab2;

import main.java.com.kupp.progLab2.pokemons.*;
import ru.ifmo.se.pokemon.Battle;

public class Lab2Battleground {
  public static void main(String[] args) {
    Battle battleground = new Battle();

    battleground.addAlly(new Shuckle("Nohtyp", 42));
    battleground.addAlly(new Diggersby("Eht", 60));
    battleground.addAlly(new Marill("Tseb", 87));

    battleground.addFoe(new Bunnelby("Avaj", 47));
    battleground.addFoe(new Azurill("Kcus", 87));
    battleground.addFoe(new Azumarill("Peed", 21));

    battleground.go();
  }
}
```

3.2 main.java.com.kupp.progLab2.moves

3.2.1 BellyDrum.java

```java
package main.java.com.kupp.progLab2.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 BellyDrum extends StatusMove {
  public BellyDrum() {
    super(Type.NORMAL, 0.0, 1.0);
  }

  @Override
  protected void applySelfEffects(Pokemon self) {
    double HP = self.getHP();
    self.setMod(Stat.HP, (int) Math.round(HP / 2));
    self.setMod(Stat.ATTACK, 6);
  }

  @Override
  protected String describe() {
    return "use belly drum";
  }
}
```

3.2.2 DoubleSlap.java

```java
package main.java.com.kupp.progLab2.moves;

import ru.ifmo.se.pokemon.PhysicalMove;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class DoubleSlap extends PhysicalMove {

  public DoubleSlap() {
    super(Type.NORMAL, 15, 0.85);
  }

  @Override
  protected String describe() {
    return "slap";
  }

  @Override
  protected double calcCriticalHit(Pokemon att, Pokemon def) {
    double TWO_HIT_RIGHT_EDGE = 0.375;
    double THREE_HIT_RIGHT_EDGE = 0.75;
    double FOUR_HIT_RIGHT_EDGE = 0.875;
    double FIVE_HIT_RIGHT_EDGE = 1.0;

    double chance = Math.random();
    int hits_count = 0;
    if (chance <= TWO_HIT_RIGHT_EDGE) {
      hits_count = 2;
    } else if (chance <= THREE_HIT_RIGHT_EDGE) {
      hits_count = 3;
    } else if (chance <= FOUR_HIT_RIGHT_EDGE) {
      hits_count = 4;
    } else if (chance <= FIVE_HIT_RIGHT_EDGE) {
      hits_count = 5;
    }

    double k = 0.0;
    for (int i = 0; i < hits_count; i++) {
      k += super.calcCriticalHit(att, def);
    }

    return k;
  }
}
```

3.2.3 Facade.java

```java
package main.java.com.kupp.progLab2.moves;

import ru.ifmo.se.pokemon.PhysicalMove;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Status;
import ru.ifmo.se.pokemon.Type;

public class Facade extends PhysicalMove {
  public Facade() {
    super(Type.NORMAL, 70, 1.0);
  }

```

```
13    @Override
14    protected String describe() {
15      return "facade";
16    }
17
18    @Override
19    protected double calcBaseDamage(Pokemon att, Pokemon def) {
20      double damage = super.calcBaseDamage(att, def);
21      Status attStatus = att.getCondition();
22      if (attStatus == Status.BURN || attStatus == Status.PARALYZE ||
            attStatus == Status.POISON) {
23        damage *= 2;
24      }
25      return damage;
26    }
27  }
```

3.2.4 HammerArm.java

```
1   package main.java.com.kupp.progLab2.moves;
2
3   import ru.ifmo.se.pokemon.PhysicalMove;
4   import ru.ifmo.se.pokemon.Pokemon;
5   import ru.ifmo.se.pokemon.Stat;
6   import ru.ifmo.se.pokemon.Type;
7
8   public class HammerArm extends PhysicalMove {
9     public HammerArm() {
10      super(Type.FIGHTING, 100, 0.9);
11    }
12
13    @Override
14    protected void applySelfEffects(Pokemon self) {
15      self.setMod(Stat.SPEED, -1);
16    }
17
18    @Override
19    protected String describe() {
20      return "applies a hammer hand";
21    }
22  }
```

3.2.5 RockThrow.java

```
1   package main.java.com.kupp.progLab2.moves;
2
3   import ru.ifmo.se.pokemon.PhysicalMove;
4   import ru.ifmo.se.pokemon.Type;
5
6   public class RockThrow extends PhysicalMove {
7     public RockThrow() {
8       super(Type.ROCK, 50, 0.9);
9     }
10
11    @Override
12    protected String describe() {
13      return "threw a rock";
14    }
15  }
```

3.2.6 SludgeBomb.java

```
1  package main.java.com.kupp.progLab2.moves;
2
3  import ru.ifmo.se.pokemon.*;
4
5  public class SludgeBomb extends SpecialMove {
6    private static Effect poisonEff = new
         Effect().chance(0.3).turns(-1).condition(Status.POISON);
7
8    public SludgeBomb() {
9      super(Type.POISON, 90, 1.0);
10   }
11
12   @Override
13   protected void applyOppEffects(Pokemon opp) {
14     opp.addEffect(poisonEff);
15   }
16
17   @Override
18   protected String describe() {
19     return "throws a sludge bomb";
20   }
21 }
```

3.2.7 Swagger.java

```
1  package main.java.com.kupp.progLab2.moves;
2
3  import ru.ifmo.se.pokemon.*;
4
5  public class Swagger extends StatusMove {
6
7    public Swagger() {
8      super(Type.NORMAL, 0.0, 1.0);
9    }
10
11   @Override
12   protected void applyOppEffects(Pokemon opp) {
13     opp.setMod(Stat.ATTACK, 2);
14     Effect.confuse(opp);
15   }
16
17   @Override
18   protected String describe() {
19     return "swagger";
20   }
21 }
```

3.2.8 SweetScent.java

```
1  package main.java.com.kupp.progLab2.moves;
2
3  import ru.ifmo.se.pokemon.Pokemon;
4  import ru.ifmo.se.pokemon.Stat;
5  import ru.ifmo.se.pokemon.StatusMove;
6  import ru.ifmo.se.pokemon.Type;
7
8  public class SweetScent extends StatusMove {
9
10   public SweetScent() {
11     super(Type.NORMAL, 0.0, 1.0);
```

```
12    }
13
14    @Override
15    protected void applyOppEffects(Pokemon opp) {
16      opp.setMod(Stat.EVASION, -1);
17    }
18
19    @Override
20    protected String describe() {
21      return "use sweet scent";
22    }
23  }
```

3.2.9 WaterGun.java

```
1  package main.java.com.kupp.progLab2.moves;
2
3  import ru.ifmo.se.pokemon.SpecialMove;
4  import ru.ifmo.se.pokemon.Type;
5
6  public class WaterGun extends SpecialMove {
7    public WaterGun() {
8      super(Type.WATER, 40, 1.0);
9    }
10
11   @Override
12   protected String describe() {
13     return "shoots with a water gun";
14   }
15 }
```

3.2.10 Withdraw.java

```
1  package main.java.com.kupp.progLab2.moves;
2
3  import ru.ifmo.se.pokemon.Pokemon;
4  import ru.ifmo.se.pokemon.Stat;
5  import ru.ifmo.se.pokemon.StatusMove;
6  import ru.ifmo.se.pokemon.Type;
7
8  public class Withdraw extends StatusMove {
9
10   public Withdraw() {
11     super(Type.WATER, 0.0, 1.0);
12   }
13
14   @Override
15   protected void applySelfEffects(Pokemon self) {
16     self.setMod(Stat.DEFENSE, 1);
17   }
18
19   @Override
20   protected String describe() {
21     return "withdraw";
22   }
23 }
```

3.3 main.java.com.kupp.progLab2.pokemons

3.3.1 Azurill.java

```
1  package main.java.com.kupp.progLab2.pokemons;
2
3  import main.java.com.kupp.progLab2.moves.Swagger;
4  import main.java.com.kupp.progLab2.moves.WaterGun;
5  import ru.ifmo.se.pokemon.Pokemon;
6  import ru.ifmo.se.pokemon.Type;
7
8  public class Azurill extends Pokemon {
9    public Azurill(String name, int level) {
10     super(name, level);
11
12     setStats(50, 20, 40, 20, 40, 20);
13
14     setType(Type.NORMAL, Type.FAIRY);
15
16     addMove(new WaterGun());
17     addMove(new Swagger());
18   }
19 }
```

3.3.2 Marill.java

```
1  package main.java.com.kupp.progLab2.pokemons;
2
3  import main.java.com.kupp.progLab2.moves.BellyDrum;
4  import ru.ifmo.se.pokemon.Type;
5
6  public class Marill extends Azurill {
7    public Marill(String name, int level) {
8      super(name, level);
9
10     setStats(70, 20, 50, 20, 50, 40);
11
12     setType(Type.WATER, Type.FAIRY);
13
14     addMove(new BellyDrum());
15   }
16 }
```

3.3.3 Azumarill.java

```
1  package main.java.com.kupp.progLab2.pokemons;
2
3  import main.java.com.kupp.progLab2.moves.WaterGun;
4
5  public class Azumarill extends Marill {
6    public Azumarill(String name, int level) {
7      super(name, level);
8
9      setStats(100, 50, 80, 60, 80, 50);
10
11     addMove(new WaterGun());
12   }
13 }
```

3.3.4 Bunnelby.java

```java
package main.java.com.kupp.progLab2.pokemons;

import main.java.com.kupp.progLab2.moves.DoubleSlap;
import main.java.com.kupp.progLab2.moves.Facade;
import main.java.com.kupp.progLab2.moves.SludgeBomb;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Bunnelby extends Pokemon {

  public Bunnelby(String name, int level) {
    super(name, level);

    setStats(38, 36, 38, 32, 36, 57);

    addType(Type.NORMAL);

    addMove(new Facade());
    addMove(new SludgeBomb());
    addMove(new DoubleSlap());
  }
}
```

3.3.5 Diggersby.java

```java
package main.java.com.kupp.progLab2.pokemons;

import main.java.com.kupp.progLab2.moves.HammerArm;
import ru.ifmo.se.pokemon.Type;

public class Diggersby extends Bunnelby {
  public Diggersby(String name, int level) {
    super(name, level);

    setStats(85, 56, 77, 50, 77, 78);

    addType(Type.GROUND);

    addMove(new HammerArm());
  }
}
```

3.3.6 Shuckle.java

```java
package main.java.com.kupp.progLab2.pokemons;

import main.java.com.kupp.progLab2.moves.Facade;
import main.java.com.kupp.progLab2.moves.RockThrow;
import main.java.com.kupp.progLab2.moves.SweetScent;
import main.java.com.kupp.progLab2.moves.Withdraw;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Shuckle extends Pokemon {
  public Shuckle(String name, int level) {
    super(name, level);

    setStats(20, 10, 230, 10, 230, 5);

    setType(Type.BUG, Type.ROCK);
```

```
17
18      addMove(new SweetScent());
19      addMove(new RockThrow());
20      addMove(new Withdraw());
21      addMove(new Facade());
22    }
23  }
```

4 Результат работы программы

```
1  Shuckle Nohtyp from the team White enters the battle!
2  Bunnelby Avaj from the team Purple enters the battle!
3  Bunnelby Avaj struggles.
4  Shuckle Nohtyp loses 6 hit points.
5  Bunnelby Avaj loses 2 hit points.
6
7  Shuckle Nohtyp facade.
8  Bunnelby Avaj loses 6 hit points.
9
10 Bunnelby Avaj throws a sludge bomb.
11 Shuckle Nohtyp loses 4 hit points.
12 Shuckle Nohtyp is poisoned
13
14 Shuckle Nohtyp threw a rock.
15 Bunnelby Avaj loses 13 hit points.
16
17 Bunnelby Avaj throws a sludge bomb.
18 Shuckle Nohtyp loses 3 hit points.
19
20 Shuckle Nohtyp withdraw.
21 Shuckle Nohtyp increases defense.
22
23 Bunnelby Avaj struggles.
24 Shuckle Nohtyp loses 4 hit points.
25 Bunnelby Avaj loses 1 hit points.
26
27 Shuckle Nohtyp threw a rock.
28 Bunnelby Avaj loses 8 hit points.
29
30 Bunnelby Avaj throws a sludge bomb.
31 Shuckle Nohtyp loses 4 hit points.
32
33 Shuckle Nohtyp withdraw.
34 Shuckle Nohtyp increases defense.
35
36 Bunnelby Avaj facade.
37 Critical hit!
38 Shuckle Nohtyp loses 7 hit points.
39
40 Shuckle Nohtyp misses
41
42 Bunnelby Avaj facade.
43 Shuckle Nohtyp loses 3 hit points.
44
45 Shuckle Nohtyp facade.
46 Critical hit!
47 Bunnelby Avaj loses 28 hit points.
48
49 Bunnelby Avaj struggles.
50 Shuckle Nohtyp loses 3 hit points.
51 Bunnelby Avaj loses 1 hit points.
```

```
52
53 Shuckle Nohtyp misses
54
55 Bunnelby Avaj struggles.
56 Shuckle Nohtyp loses 5 hit points.
57 Bunnelby Avaj loses 1 hit points.
58
59 Shuckle Nohtyp facade.
60 Bunnelby Avaj loses 15 hit points.
61
62 Bunnelby Avaj struggles.
63 Critical hit!
64 Shuckle Nohtyp loses 9 hit points.
65 Bunnelby Avaj loses 2 hit points.
66
67 Shuckle Nohtyp facade.
68 Bunnelby Avaj loses 14 hit points.
69
70 Bunnelby Avaj throws a sludge bomb.
71 Shuckle Nohtyp loses 3 hit points.
72
73 Shuckle Nohtyp facade.
74 Bunnelby Avaj loses 10 hit points.
75 Bunnelby Avaj faints.
76 Azurill Kcus from the team Purple enters the battle!
77 Azurill Kcus struggles.
78 Shuckle Nohtyp loses 6 hit points.
79 Azurill Kcus loses 2 hit points.
80
81 Shuckle Nohtyp facade.
82 Azurill Kcus loses 11 hit points.
83
84 Azurill Kcus shoots with a water gun.
85 Shuckle Nohtyp loses 18 hit points.
86
87 Shuckle Nohtyp use sweet scent.
88 Azurill Kcus decreases evasion.
89
90 Azurill Kcus struggles.
91 Shuckle Nohtyp loses 4 hit points.
92 Azurill Kcus loses 1 hit points.
93 Shuckle Nohtyp faints.
94 Diggersby Eht from the team White enters the battle!
95 Diggersby Eht misses
96
97 Azurill Kcus struggles.
98 Diggersby Eht loses 15 hit points.
99 Azurill Kcus loses 4 hit points.
100
101 Diggersby Eht throws a sludge bomb.
102 Critical hit!
103 Azurill Kcus loses 84 hit points.
104
105 Azurill Kcus shoots with a water gun.
106 Critical hit!
107 Diggersby Eht loses 26 hit points.
108
109 Diggersby Eht misses
110
111 Azurill Kcus swagger.
112 Diggersby Eht increases attack.
```

```
113
114 Diggersby Eht misses
115
116 Azurill Kcus struggles.
117 Critical hit!
118 Diggersby Eht loses 19 hit points.
119 Azurill Kcus loses 5 hit points.
120
121 Diggersby Eht slap.
122 Critical hit!
123 Critical hit!
124 Azurill Kcus loses 52 hit points.
125
126 Azurill Kcus swagger.
127 Diggersby Eht increases attack.
128
129 Diggersby Eht facade.
130 Azurill Kcus loses 63 hit points.
131 Azurill Kcus faints.
132 Azumarill Peed from the team Purple enters the battle!
133 Diggersby Eht hits himself in confusion.
134 Diggersby Eht loses 53 hit points.
135
136 Azumarill Peed swagger.
137 Diggersby Eht increases attack.
138
139 Diggersby Eht throws a sludge bomb.
140 Azumarill Peed loses 76 hit points.
141 Azumarill Peed is poisoned
142
143 Azumarill Peed swagger.
144 Diggersby Eht increases attack.
145
146 Diggersby Eht hits himself in confusion.
147 Diggersby Eht loses 55 hit points.
148
149 Azumarill Peed shoots with a water gun.
150 Diggersby Eht loses 6 hit points.
151
152 Diggersby Eht facade.
153 Critical hit!
154 Azumarill Peed loses 452 hit points.
155 Azumarill Peed faints.
156 Team Purple loses its last Pokemon.
157 The team White wins the battle!
```

5 Вывод

В ходе этой лабораторной работы я поработал с документацией к классу Pokemon и реализовал на его основе заданных покемонов и заданные атаки, как следствие поработал с ООП в Java.