



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

Выполнил:
Студент группы Р3113
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	14

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.

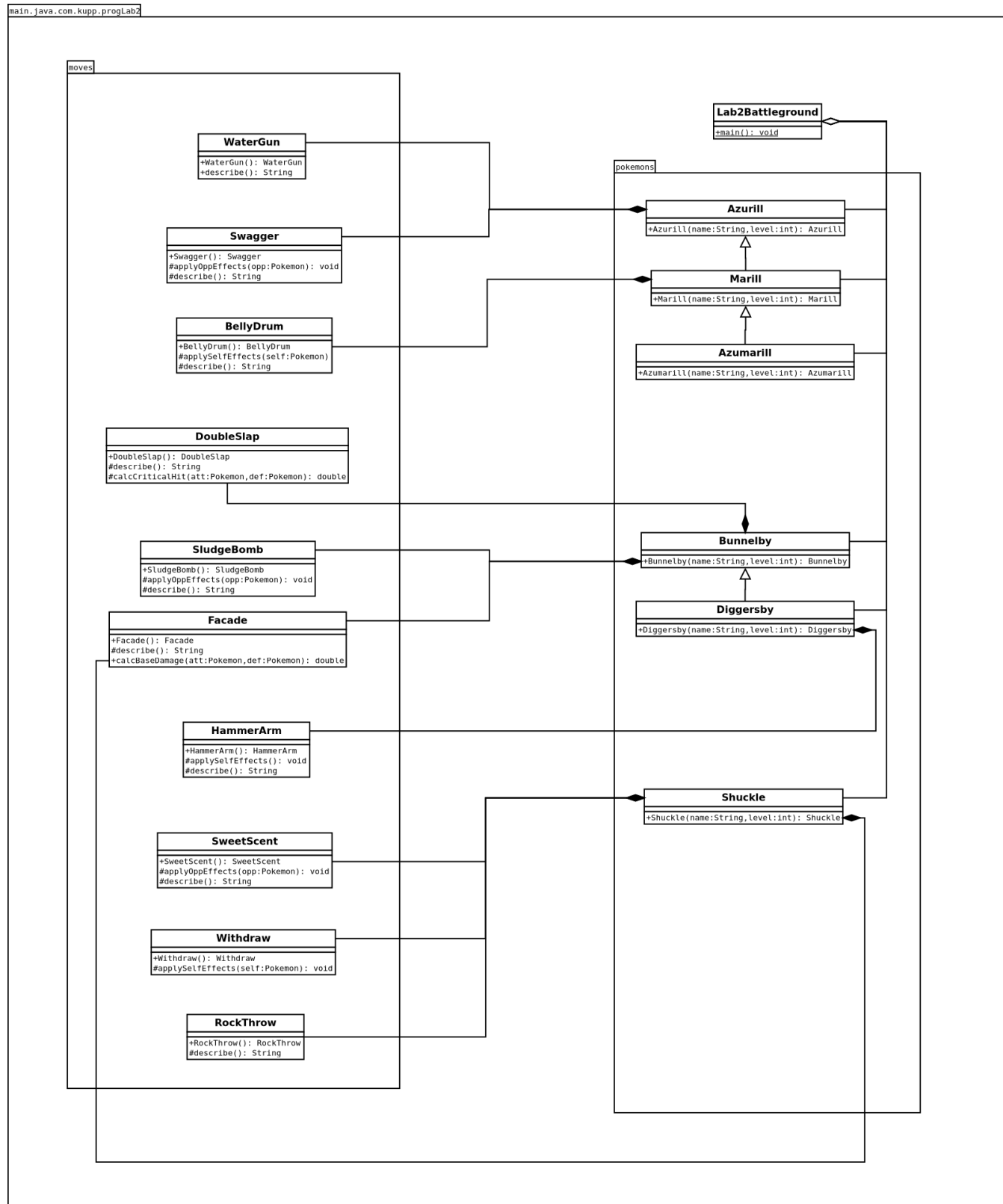
Используя класс симуляции боя **Battle**, создать 2 команды покемонов (каждый покемон должен иметь имя) и запустить бой.

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/](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

```
1 package main.java.com.kupp.progLab2;
2
3 import main.java.com.kupp.progLab2.pokemons.*;
4 import ru.ifmo.se.pokemon.Battle;
5
6 public class Lab2Battleground {
7     public static void main(String[] args) {
8         Battle battleground = new Battle();
9
10        battleground.addAlly(new Shuckle("Положить болтналибу ", 42));
11        battleground.addAlly(new Diggersby("Прокрастинировать", 60));
12        battleground.addAlly(new Marill("Молятся солиду", 87));
13
14        battleground.addFoe(new Bunnelby("Юзать солид", 47));
15        battleground.addFoe(new Azurill("Сделать нормлибу ", 87));
16        battleground.addFoe(new Azumarill("Мы молодцы", 21));
17
18        battleground.go();
19    }
20 }
```

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

3.2.1 BellyDrum.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 BellyDrum extends StatusMove {
9     public BellyDrum() {
10         super(Type.NORMAL, 0.0, 1.0);
11     }
12
13     @Override
14     protected void applySelfEffects(Pokemon self) {
15         double HP = self.getHP();
16         self.setMod(Stat.HP, (int) Math.round(HP / 2));
17         self.setMod(Stat.ATTACK, 6);
18     }
19
20     @Override
21     protected String describe() {
22         return "use belly drum";
23     }
24 }
```

3.2.2 DoubleSlap.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.Type;
6
7 public class DoubleSlap extends PhysicalMove {
8
9     public DoubleSlap() {
10         super(Type.NORMAL, 15, 0.85);
11     }
12
13     @Override
14     protected String describe() {
15         return "slap";
16     }
17
18     @Override
19     protected double calcCriticalHit(Pokemon att, Pokemon def) {
20         double TWO_HIT_RIGHT_EDGE = 0.375;
21         double THREE_HIT_RIGHT_EDGE = 0.75;
22         double FOUR_HIT_RIGHT_EDGE = 0.875;
23         double FIVE_HIT_RIGHT_EDGE = 1.0;
24
25         double chance = Math.random();
26         int hits_count = 0;
27         if (chance <= TWO_HIT_RIGHT_EDGE) {
28             hits_count = 2;
29         } else if (chance <= THREE_HIT_RIGHT_EDGE) {
30             hits_count = 3;
31         } else if (chance <= FOUR_HIT_RIGHT_EDGE) {
32             hits_count = 4;
33         } else if (chance <= FIVE_HIT_RIGHT_EDGE) {
34             hits_count = 5;
35         }
36
37         double k = 0.0;
38         for (int i = 0; i < hits_count; i++) {
39             k += super.calcCriticalHit(att, def);
40         }
41
42         return k;
43     }
44 }
```

3.2.3 Facade.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.Status;
6 import ru.ifmo.se.pokemon.Type;
7
8 public class Facade extends PhysicalMove {
9     public Facade() {
10         super(Type.NORMAL, 70, 1.0);
11     }
12 }
```

```

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 ||
23             attStatus == Status.POISON) {
24             damage *= 2;
25         }
26         return damage;
27     }

```

3.2.4 HammerArm.java

```

1 package main.java.com.kupp.progLab2.moves;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class HammerArm extends PhysicalMove {
6     private static Effect decSp = new
7         Effect().chance(1.0).turns(-1).stat(Stat.SPEED, -1);
8
9     public HammerArm() {
10         super(Type.FIGHTING, 100, 0.9);
11     }
12
13     @Override
14     protected void applySelfEffects(Pokemon self) {
15         self.addEffect(decSp);
16     }
17
18     @Override
19     protected String describe() {
20         return "applies a hammer hand";
21     }

```

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
7         Effect().chance(0.3).turns(-1).condition(Status.POISON);
8
9     public SludgeBomb() {
10         super(Type.POISON, 90, 1.0);
11     }
12
13     @Override
14     protected void applyOppEffects(Pokemon opp) {
15         opp.addEffect(poisonEff);
16     }
17
18     @Override
19     protected String describe() {
20         return "throws a sludge bomb";
21     }
22 }
```

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     private static Effect inc2Att = new
7         Effect().chance(1.0).turns(-1).stat(Stat.ATTACK, 2);
8
9     public Swagger() {
10         super(Type.NORMAL, 0.0, 1.0);
11     }
12
13     @Override
14     protected void applyOppEffects(Pokemon opp) {
15         opp.addEffect(inc2Att);
16         Effect.confuse(opp);
17     }
18
19     @Override
20     protected String describe() {
21         return "swagger";
22     }
23 }
```

3.2.8 SweetScent.java

```
1 package main.java.com.kupp.progLab2.moves;
2
3 import ru.ifmo.se.pokemon.*;
4
5 public class SweetScent extends StatusMove {
6     private static Effect decEv = new
7         Effect().chance(1.0).turns(-1).stat(Stat.EVASION, -1);
8
9     public SweetScent() {
```



```

9      super(Type.NORMAL, 0.0, 1.0);
10  }
11
12  @Override
13  protected void applyOppEffects(Pokemon opp) {
14      opp.addEffect(decEv);
15  }
16
17  @Override
18  protected String describe() {
19      return "use sweet scent";
20  }
21  }

```

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.*;
4
5  public class Withdraw extends StatusMove {
6      private static Effect incDef;
7
8      public Withdraw() {
9          super(Type.WATER, 0.0, 1.0);
10         incDef = new Effect().chance(1.0).turns(-1).stat(Stat.DEFENSE, 1);
11     }
12
13  @Override
14  protected void applySelfEffects(Pokemon self) {
15      self.addEffect(incDef);
16  }
17  }

```

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

```
1 package main.java.com.kupp.progLab2.pokemons;
2
3 import main.java.com.kupp.progLab2.moves.DoubleSlap;
4 import main.java.com.kupp.progLab2.moves.Facade;
5 import main.java.com.kupp.progLab2.moves.SludgeBomb;
6 import ru.ifmo.se.pokemon.Pokemon;
7 import ru.ifmo.se.pokemon.Type;
8
9 public class Bunnelby extends Pokemon {
10
11     public Bunnelby(String name, int level) {
12         super(name, level);
13
14         setStats(38, 36, 38, 32, 36, 57);
15
16         addType(Type.NORMAL);
17
18         addMove(new Facade());
19         addMove(new SludgeBomb());
20         addMove(new DoubleSlap());
21     }
22 }
```

3.3.5 Diggersby.java

```
1 package main.java.com.kupp.progLab2.pokemons;
2
3 import main.java.com.kupp.progLab2.moves.HammerArm;
4 import ru.ifmo.se.pokemon.Type;
5
6 public class Diggersby extends Bunnelby {
7     public Diggersby(String name, int level) {
8         super(name, level);
9
10         setStats(85, 56, 77, 50, 77, 78);
11
12         addType(Type.GROUND);
13
14         addMove(new HammerArm());
15     }
16 }
```

3.3.6 Shuckle.java

```
1 package main.java.com.kupp.progLab2.pokemons;
2
3 import main.java.com.kupp.progLab2.moves.Facade;
4 import main.java.com.kupp.progLab2.moves.RockThrow;
5 import main.java.com.kupp.progLab2.moves.SweetScent;
6 import main.java.com.kupp.progLab2.moves.Withdraw;
7 import ru.ifmo.se.pokemon.Pokemon;
8 import ru.ifmo.se.pokemon.Type;
9
10 public class Shuckle extends Pokemon {
11     public Shuckle(String name, int level) {
12         super(name, level);
13
14         setStats(20, 10, 230, 10, 230, 5);
15
16         setType(Type.BUG, Type.ROCK);
17     }
18 }
```

```

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

```

52 Shuckle Положитьболтналибу loses 3 hit points.
 53
 54 Shuckle Положитьболтналибу attacks.
 55
 56 Bunnelby Юзатьсясолид throws a sludge bomb.
 57 Shuckle Положитьболтналибу loses 4 hit points.
 58
 59 Shuckle Положитьболтналибу attacks.
 60
 61 Bunnelby Юзатьсясолид slap.
 62 Critical hit!
 63 Shuckle Положитьболтналибу loses 5 hit points.
 64
 65 Shuckle Положитьболтналибу threw a rock.
 66 Bunnelby Юзатьсясолид loses 10 hit points.
 67
 68 Bunnelby Юзатьсясолид throws a sludge bomb.
 69 Shuckle Положитьболтналибу loses 3 hit points.
 70
 71 Shuckle Положитьболтналибу use sweet scent.
 72
 73 Bunnelby Юзатьсясолид slap.
 74 Shuckle Положитьболтналибу loses 4 hit points.
 75
 76 Shuckle Положитьболтналибу misses
 77
 78 Bunnelby Юзатьсясолид throws a sludge bomb.
 79 Shuckle Положитьболтналибу loses 5 hit points.
 80
 81 Shuckle Положитьболтналибу threw a rock.
 82 Bunnelby Юзатьсясолид loses 10 hit points.
 83
 84 Bunnelby Юзатьсясолид facade.
 85 Shuckle Положитьболтналибу loses 4 hit points.
 86
 87 Shuckle Положитьболтналибу misses
 88
 89 Bunnelby Юзатьсясолид slap.
 90 Shuckle Положитьболтналибу loses 5 hit points.
 91
 92 Shuckle Положитьболтналибу misses
 93
 94 Bunnelby Юзатьсясолид facade.
 95 Shuckle Положитьболтналибу loses 3 hit points.
 96
 97 Shuckle Положитьболтналибу misses
 98
 99 Bunnelby Юзатьсясолид facade.
 100 Shuckle Положитьболтналибу loses 4 hit points.
 101 Shuckle Положитьболтналибу faints.
 102 Diggersby Прокрастинировать from the team Striped enters the battle!
 103 Diggersby Прокрастинировать misses
 104
 105 Bunnelby Юзатьсясолид throws a sludge bomb.
 106 Diggersby Прокрастинировать loses 11 hit points.
 107
 108 Diggersby Прокрастинировать facade.
 109 Bunnelby Юзатьсясолид loses 45 hit points.
 110
 111 Bunnelby Юзатьсясолид slap.
 112 Diggersby Прокрастинировать loses 19 hit points.

113
114 Diggersby Прокрастинировать applies a hammer hand.
115 Bunnelby Юзатьсясолид loses 81 hit points.
116 Bunnelby Юзатьсясолид faints.
117 Azurill Сделатьнормлибу from the team White enters the battle!
118 Diggersby Прокрастинировать throws a sludge bomb.
119 Critical hit!
120 Azurill Сделатьнормлибу loses 99 hit points.
121 Azurill Сделатьнормлибу is poisoned
122
123 Azurill Сделатьнормлибу swagger.
124
125 Diggersby Прокрастинировать applies a hammer hand.
126 Azurill Сделатьнормлибу loses 58 hit points.
127
128 Azurill Сделатьнормлибу shoots with a water gun.
129 Diggersby Прокрастинировать loses 8 hit points.
130
131 Diggersby Прокрастинировать slap.
132 Critical hit!
133 Critical hit!
134 Azurill Сделатьнормлибу loses 72 hit points.
135 Azurill Сделатьнормлибу faints.
136 Azumarill Мымолодцы from the team White enters the battle!
137 Diggersby Прокрастинировать hits himself in confusion.
138 Diggersby Прокрастинировать loses 29 hit points.
139
140 Azumarill Мымолодцы shoots with a water gun.
141 Diggersby Прокрастинировать loses 5 hit points.
142
143 Diggersby Прокрастинировать slap.
144 Critical hit!
145 Critical hit!
146 Critical hit!
147 Azumarill Мымолодцы loses 186 hit points.
148 Azumarill Мымолодцы faints.
149 Team White loses its last Pokemon.
150 The team Striped 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.