

Санкт-Петербургский Национальный Исследовательский
Университет Информационных Технологий, Механики и Оптики
Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа №2
по дисциплине
“Программирование”

Вариант:311698
Выполнил
Студент группы Р3116
Чэнь Хаолин : 407960

Санкт-Петербург 2024 г.

Содержание

Текст задания

UML

Исходный код программы

- **package Fight**

Main

- **package Move**

Facade

FocusBlast

IcicleSpear

Liquidation

MudBomb

Rest

RockPolish

SlackOff

- **package Pokemon**

Blissey

Chansey

Corsola

Happiny

Slowbro

Slowpke

Результат работы программы

Выводы по работе

Санкт-Петербург 2024 г.

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)

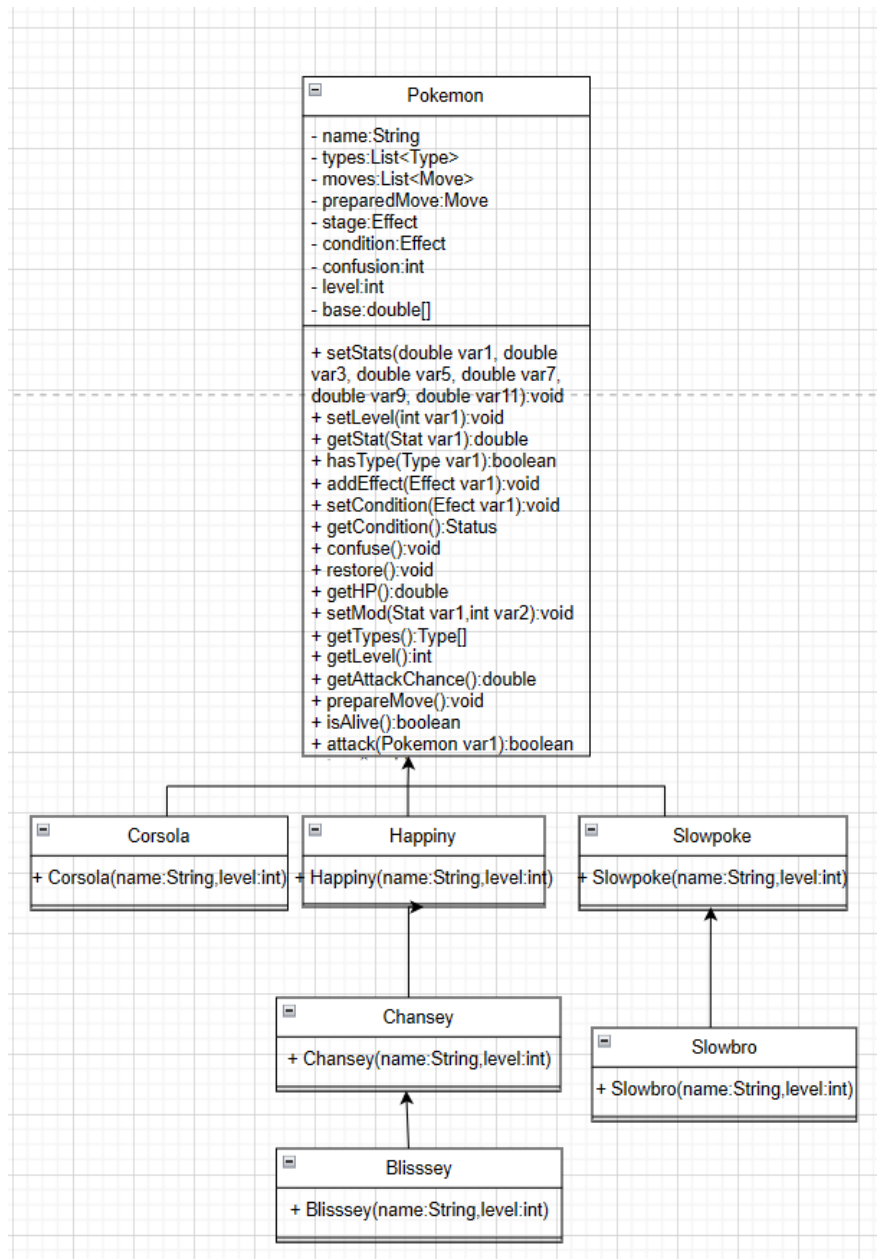
Классы покемонов должны наследоваться в соответствии с цепочкой эволюции покемонов. На основе базовых классов `PhysicalMove`, `SpecialMove` и `StatusMove` реализовать свои классы для заданных видов атак. Все разработанные классы, не имеющие наследников, должны быть реализованы таким образом, чтобы от них нельзя было наследоваться.

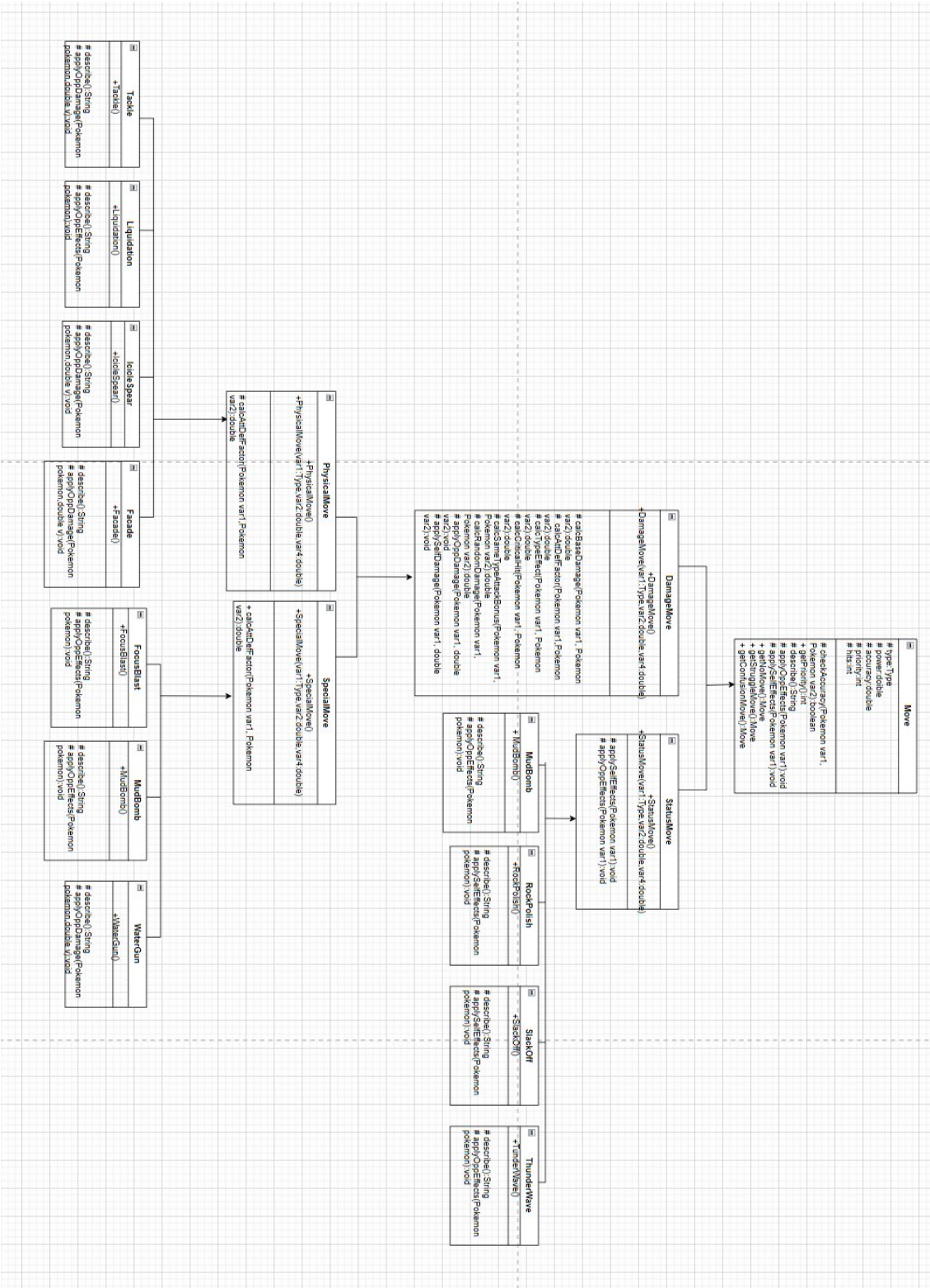
Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атаки. Назначить каждому виду покемонов атаки в соответствии с вариантом. Уровень покемона выбирается минимально необходимым для всех реализованных атак.

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

<p>Corsola</p>  <p>Атаки: Rock Polish Liquidation Icicle Spear Tackle</p>	<p>Slowpoke</p>  <p>Атаки: Rest Slack Off Water Gun</p>	<p>Slowbro</p>  <p>Атаки: Rest Slack Off Water Gun Focus Blast</p>	<p>Happiny</p>  <p>Атаки: Rest Facade</p>	<p>Chansey</p>  <p>Атаки: Rest Facade Mud Bomb</p>	<p>Blissey</p>  <p>Атаки: Rest Facade Mud Bomb Thunder Wave</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

UML





Исходный код программы

package Fight

Main

```
1. package Fight;
2.
3. import MyPokemon.*;
4. import ru.ifmo.se.pokemon.Battle;
5.
6. public class Main {
7.     public static void main(String[] args) {
8.         Battle battle = new Battle();
9.         battle.addAlly(new Corsola("t1-1",30));
10.        battle.addAlly(new Slowpoke("t1-2",70));
11.        battle.addAlly(new Slowbro("t1-3",100));
12.        battle.addFoe(new Happiny("t2-1",30));
13.        battle.addFoe(new Chansey("t2-2",70));
14.        battle.addFoe(new Blissey("t2-3",100));
15.        battle.go();
16.    }
17. }
```

package Move

Facade

```
1. package Move;
2.
3. import ru.ifmo.se.pokemon.*;
4.
5. public class Facade extends PhysicalMove {
6.     public Facade() {
7.         super(Type.NORMAL, 70, 100);
8.     }
9.
10.    @Override
11.    protected void applyOppDamage(Pokemon pokemon, double v) {
12.        pokemon.setMod(Stat.HP, 70);
13.    }
14. }
```

```

15.     @Override
16.     protected String describe() {
17.         return "used " + this.getClass().getSimpleName();
18.     }
19. }

```

FocusBlast

```

1. package Move;
2.
3. import ru.ifmo.se.pokemon.*;
4.
5. public class FocusBlast extends SpecialMove {
6.     public FocusBlast() {
7.         super(Type.FIGHTING, 120, 70);
8.     }
9.
10.    @Override
11.    protected void applyOppEffects(Pokemon pokemon) {
12.        if (Math.random() < 0.1) {
13.            pokemon.addEffect(new Effect().stat(Stat.DEFENSE, -1));
14.        }
15.    }
16.
17.    @Override
18.    protected String describe() {
19.        return "used " + this.getClass().getSimpleName();
20.    }
21. }

```

IcicleSpear

```

1. package Move;
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 IcicleSpear extends PhysicalMove {
9.     public IcicleSpear() {
10.        super(Type.ICE, 25, 100);
11.    }

```

```

12.
13.     @Override
14.     protected void applyOppDamage(Pokemon pokemon, double v) {
15.         if (Math.random() <= 0.375) {
16.             pokemon.setMod(Stat.HP, 50);
17.         } else if (Math.random() <= 0.75) {
18.             pokemon.setMod(Stat.HP, 75);
19.         } else if (Math.random() <= 0.875) {
20.             pokemon.setMod(Stat.HP, 100);
21.         } else {
22.             pokemon.setMod(Stat.HP, 125);
23.         }
24.     }
25.
26.     @Override
27.     protected String describe() {
28.         return "used " + this.getClass().getSimpleName();
29.     }
30. }

```

Liquidation

```

1. package Move;
2.
3. import ru.ifmo.se.pokemon.*;
4.
5. public class Liquidation extends PhysicalMove {
6.     public Liquidation() {
7.         super(Type.WATER, 85, 100);
8.     }
9.
10.    @Override
11.    protected void applyOppEffects(Pokemon pokemon) {
12.        if (Math.random() <= 0.2) {
13.            pokemon.addEffect(new Effect().stat(Stat.DEFENSE, -1));
14.        }
15.    }
16.
17.    @Override
18.    protected String describe() {
19.        return "used " + this.getClass().getSimpleName();
20.    }
21. }

```


MudBomb

```
1. package Move;
2.
3. import ru.ifmo.se.pokemon.*;
4.
5. public class MudBomb extends SpecialMove {
6.     public MudBomb() {
7.         super(Type.GROUND, 65, 85);
8.     }
9.
10.    @Override
11.    protected void applyOppEffects(Pokemon pokemon) {
12.        if (Math.random() <= 0.3){
13.            pokemon.addEffect(new Effect().stat(Stat.ACCURACY, -1));
14.        }
15.    }
16.
17.    @Override
18.    protected String describe() {
19.        return "used " + this.getClass().getSimpleName();
20.    }
21. }
```

Rest

```
1. package Move;
2.
3. import ru.ifmo.se.pokemon.Effect;
4. import ru.ifmo.se.pokemon.Pokemon;
5. import ru.ifmo.se.pokemon.StatusMove;
6. import ru.ifmo.se.pokemon.Type;
7.
8. public class Rest extends StatusMove {
9.     public Rest(){
10.         super(Type.PSYCHIC, 0, 0);
11.     }
12.
13.    @Override
14.    protected void applyOppEffects(Pokemon pokemon) {
15.        Effect.sleep(pokemon);
16.    }
17. }
```

```

18.     @Override
19.     protected String describe() {
20.         return "used " + this.getClass().getSimpleName();
21.     }
22. }

```

RockPolish

```

1. package Move;
2.
3. import ru.ifmo.se.pokemon.*;
4.
5. public class RockPolish extends StatusMove {
6.     public RockPolish() {
7.         super(Type.ROCK, 0, 0);
8.     }
9.
10.    @Override
11.    protected void applySelfEffects(Pokemon pokemon) {
12.        pokemon.addEffect(new Effect().stat(Stat.SPEED, 2));
13.    }
14.
15.    @Override
16.    protected String describe() {
17.        return "used " + this.getClass().getName();
18.    }
19. }

```

SlackOff

```

1. package Move;
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 SlackOff extends StatusMove {
9.     public SlackOff() {
10.        super(Type.NORMAL, 0, 0);
11.    }
12.
13.    @Override

```

```

14.     protected void applySelfEffects(Pokemon pokemon) {
15.         pokemon.setMod(Stat.HP, (int) (pokemon.getHP() * (Math.random() * 0.
            5 + 1)));
16.     }
17.
18.     @Override
19.     protected String describe() {
20.         return "used " + this.getClass().getSimpleName();
21.     }
22. }

```

Tackle

```

1. package Move;
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 Tackle extends PhysicalMove {
9.     public Tackle() {
10.         super(Type.NORMAL, 40, 100);
11.     }
12.
13.     @Override
14.     protected void applyOppDamage(Pokemon pokemon, double v) {
15.         pokemon.setMod(Stat.HP, 50);
16.     }
17.
18.     @Override
19.     protected String describe() {
20.         return "used " + this.getClass().getSimpleName();
21.     }
22. }

```

ThunderWave

```

1. package Move;
2.
3. import ru.ifmo.se.pokemon.Effect;
4. import ru.ifmo.se.pokemon.Pokemon;
5. import ru.ifmo.se.pokemon.StatusMove;

```

```

6. import ru.ifmo.se.pokemon.Type;
7.
8. public class ThunderWave extends StatusMove {
9.     public ThunderWave(){
10.         super(Type.ELECTRIC,0,90);
11.     }
12.
13.     @Override
14.     protected void applyOppEffects(Pokemon pokemon) {
15.         Effect.paralyze(pokemon);
16.     }
17.
18.     @Override
19.     protected String describe() {
20.         return "used " + this.getClass().getSimpleName();
21.     }
22. }

```

WaterGun

```

1. package Move;
2.
3. import ru.ifmo.se.pokemon.Pokemon;
4. import ru.ifmo.se.pokemon.SpecialMove;
5. import ru.ifmo.se.pokemon.Stat;
6. import ru.ifmo.se.pokemon.Type;
7.
8. public class WaterGun extends SpecialMove {
9.     public WaterGun(){
10.         super(Type.WATER,40,100);
11.     }
12.
13.     @Override
14.     protected void applyOppDamage(Pokemon pokemon, double v) {
15.         pokemon.setMod(Stat.HP,1);
16.     }
17.
18.     @Override
19.     protected String describe() {
20.         return "used " + this.getClass().getSimpleName();
21.     }
22. }

```

package Pokemon

Blissey

```
1. package MyPokemon;
2.
3. import Move.ThunderWave;
4. import ru.ifmo.se.pokemon.Type;
5.
6. public class Blissey extends Chansey {
7.     public Blissey(String name, int level) {
8.         super(name, level);
9.         setType(Type.NORMAL);
10.        setStats(255, 10, 10, 75, 135, 55);
11.        ThunderWave move4 = new ThunderWave();
12.        addMove(move4);
13.    }
14. }
```

Chansey

```
1. package MyPokemon;
2.
3. import Move.MudBomb;
4. import ru.ifmo.se.pokemon.Type;
5.
6. public class Chansey extends Happiny {
7.     public Chansey(String name, int level) {
8.         super(name, level);
9.         setType(Type.NORMAL);
10.        setStats(250, 5, 5, 35, 105, 50);
11.        MudBomb move3 = new MudBomb();
12.        addMove(move3);
13.    }
14. }
```

Corsola

```
1. package MyPokemon;
2.
3. import Move.IcicleSpear;
4. import Move.Liquidation;
5. import Move.RockPolish;
```

```

6. import Move.Tackle;
7. import ru.ifmo.se.pokemon.Pokemon;
8. import ru.ifmo.se.pokemon.Type;
9.
10. public class Corsola extends Pokemon {
11.     public Corsola(String name, int level) {
12.         super(name, level);
13.         setType(Type.WATER, Type.ROCK);
14.         setStats(65, 55, 95, 65, 95, 35);
15.         RockPolish move1 = new RockPolish();
16.         Liquidation move2 = new Liquidation();
17.         IcicleSpear move3 = new IcicleSpear();
18.         Tackle move4 = new Tackle();
19.         setMove(move1, move2, move3, move4);
20.     }
21. }

```

Happiny

```

1. package MyPokemon;
2.
3. import Move.Facade;
4. import Move.Rest;
5. import ru.ifmo.se.pokemon.Pokemon;
6. import ru.ifmo.se.pokemon.Type;
7.
8. public class Happiny extends Pokemon {
9.     public Happiny(String name, int level) {
10.         super(name, level);
11.         setType(Type.NORMAL);
12.         setStats(100, 5, 5, 15, 65, 30);
13.         Rest move1 = new Rest();
14.         Facade move2 = new Facade();
15.         setMove(move1, move2);
16.     }
17. }

```

Slowbro

```

1. package MyPokemon;
2.
3. import Move.FocusBlast;
4. import Move.Rest;

```

```

5. import Move.SlackOff;
6. import Move.WaterGun;
7. import ru.ifmo.se.pokemon.Type;
8.
9. public class Slowbro extends Slowpoke {
10.
11.     public Slowbro(String name, int level) {
12.         super(name, level);
13.         setType(Type.WATER, Type.PSYCHIC);
14.         setStats(90, 65, 65, 40, 40, 15);
15.         FocusBlast move4 = new FocusBlast();
16.         addMove(move4);
17.     }
18. }

```

Slowpoke

```

1. package MyPokemon;
2.
3. import Move.Rest;
4. import Move.SlackOff;
5. import Move.WaterGun;
6. import ru.ifmo.se.pokemon.Pokemon;
7. import ru.ifmo.se.pokemon.Type;
8.
9. public class Slowpoke extends Pokemon {
10.     public Slowpoke(String name, int level) {
11.         super(name, level);
12.         setType(Type.WATER, Type.PSYCHIC);
13.         setStats(90, 65, 65, 40, 40, 15);
14.         Rest move1 = new Rest();
15.         SlackOff move2 = new SlackOff();
16.         WaterGun move3 = new WaterGun();
17.         setMove(move1, move2, move3);
18.     }
19. }

```

Результат работы программы

1. Corsola t1-1 from the team White enters the battle!
2. Happiny t2-1 from the team Greren enters the battle!
3. Corsola t1-1 used IcicleSpear.
4. Critical hit!
5. Happiny t2-1 loses 100 hit points.
- 6.
7. Happiny t2-1 used Facade.
8. Corsola t1-1 loses 70 hit points.
- 9.
10. Corsola t1-1 used IcicleSpear.
11. Happiny t2-1 loses 50 hit points.
12. Happiny t2-1 faints.
13. Chansey t2-2 from the team Greren enters the battle!
14. Chansey t2-2 used Facade.
15. Corsola t1-1 loses 70 hit points.
16. Corsola t1-1 faints.
17. Slowpoke t1-2 from the team White enters the battle!
18. Chansey t2-2 used Facade.
19. Slowpoke t1-2 loses 70 hit points.
- 20.
21. Slowpoke t1-2 misses
- 22.
23. Chansey t2-2 used Facade.
24. Slowpoke t1-2 loses 70 hit points.
- 25.
26. Slowpoke t1-2 used WaterGun.
27. Chansey t2-2 loses 1 hit points.
- 28.
29. Chansey t2-2 used MudBomb.
30. Slowpoke t1-2 loses 26 hit points.
- 31.
32. Slowpoke t1-2 misses
- 33.
34. Chansey t2-2 used MudBomb.
35. Slowpoke t1-2 loses 15 hit points.
- 36.
37. Slowpoke t1-2 misses
- 38.
39. Chansey t2-2 misses
- 40.

Санкт-Петербург 2024 г.

41. Slowpoke t1-2 misses
42.
43. Chansey t2-2 used MudBomb.
44. Slowpoke t1-2 loses 25 hit points.
45.
46. Slowpoke t1-2 misses
47.
48. Chansey t2-2 misses
49.
50. Slowpoke t1-2 misses
51.
52. Chansey t2-2 used MudBomb.
53. Critical hit!
54. Slowpoke t1-2 loses 42 hit points.
55. Slowpoke t1-2 faints.
56. Slowbro t1-3 from the team White enters the battle!
57. Chansey t2-2 used MudBomb.
58. Slowbro t1-3 loses 16 hit points.
59.
60. Slowbro t1-3 misses
61.
62. Chansey t2-2 misses
63.
64. Slowbro t1-3 misses
65.
66. Chansey t2-2 used MudBomb.
67. Slowbro t1-3 loses 15 hit points.
68.
69. Slowbro t1-3 used FocusBlast.
70. Chansey t2-2 loses 68 hit points.
71.
72. Chansey t2-2 used MudBomb.
73. Critical hit!
74. Slowbro t1-3 loses 27 hit points.
75.
76. Slowbro t1-3 used WaterGun.
77. Chansey t2-2 loses 1 hit points.
78.
79. Chansey t2-2 used MudBomb.
80. Critical hit!
81. Slowbro t1-3 loses 24 hit points.
82.
83. Slowbro t1-3 misses

84. Chansey t2-2 used MudBomb.
85. Slowbro t1-3 loses 17 hit points.
86.
87. Slowbro t1-3 misses
88.
89. Chansey t2-2 used MudBomb.
90. Slowbro t1-3 loses 12 hit points.
91.
92. Slowbro t1-3 used WaterGun.
93. Chansey t2-2 loses 1 hit points.
94.
95. Chansey t2-2 used MudBomb.
96. Critical hit!
97. Slowbro t1-3 loses 26 hit points.
98.
99. Slowbro t1-3 misses
100.
101. Chansey t2-2 used MudBomb.
102. Slowbro t1-3 loses 18 hit points.
103.
104. Slowbro t1-3 misses
105.
106. Chansey t2-2 used Facade.
107. Slowbro t1-3 loses 70 hit points.
108.
109. Slowbro t1-3 misses
110.
111. Chansey t2-2 used MudBomb.
112. Critical hit!
113. Slowbro t1-3 loses 39 hit points.
114.
115. Slowbro t1-3 misses
116.
117. Chansey t2-2 used MudBomb.
118. Slowbro t1-3 loses 15 hit points.
119.
120. Slowbro t1-3 misses
121.
122. Chansey t2-2 used Facade.
123. Slowbro t1-3 loses 70 hit points.
124. Slowbro t1-3 faints.
125. Team White loses its last Pokemon.
126. The team Greren wins the battle!

Выводы по работе

В ходе данной работы я научился подключать внешние jar файлы и использовать их.

Научился работать с документацией, изучил объектно-ориентированный подход на языке Java. Научился работать с классами, конструкторами, полями и модификаторами доступа.