

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ  
ТЕХНИКИ**

**ЛАБОРАТОРНАЯ №2**

ПО ДИСЦИПЛИНЕ

«ПРОГРАММИРОВАНИЕ»

ВАРИАНТ №311909

**Выполнила:**

Студентка группы Р3119

Петрова Анастасия Александровна

**Преподаватель:**

Рыбаков Степан Дмитриевич

Санкт-Петербург  
2022

## Оглавление

<b>Задание.....</b>	<b>3</b>
<b>Исходный код программы .....</b>	<b>4</b>
<b>Компиляция и разворачивание программы.....</b>	<b>9</b>
<b>Диаграмма классов.....</b>	<b>10</b>
<b>Выводы .....</b>	<b>10</b>

## Задание

1. Ознакомиться с [документацией](#), обращая особое внимание на классы **Pokemon** и **Move**. При дальнейшем выполнении лабораторной работы читать документацию еще несколько раз.
2. Скачать файл `Pokemon.jar`. Его необходимо будет использовать как для компиляции, так и для запуска программы. Распаковывать его не надо! Нужно научиться подключать внешние `jar`-файлы к своей программе.
3. Написать минимально работающую программу и посмотреть как она работает.

```
Battle b = new Battle();
Pokemon p1 = new Pokemon("Чужой", 1);
Pokemon p2 = new Pokemon("Хищник", 1);
b.addAlly(p1);
b.addFoe(p2);
b.go();
```
4. Создать один из классов покемонов для своего варианта. Класс должен наследоваться от базового класса **Pokemon**. В конструкторе нужно будет задать типы покемона и его базовые характеристики. После этого попробуйте добавить покемона в сражение.
5. Создать один из классов атак для своего варианта (лучше всего начать с физической или специальной атаки). Класс должен наследоваться от класса **PhysicalMove** или **SpecialMove**. В конструкторе нужно будет задать тип атаки, ее силу и точность. После этого добавить атаку покемону и проверить ее действие в сражении. Не забудьте переопределить метод **describe**, чтобы выводилось нужное сообщение.
6. Если действие атаки отличается от стандартного, например, покемон не промахивается, либо атакующий покемон также получает повреждение, то в классе атаки нужно дополнительно переопределить соответствующие методы (см. документацию). При реализации атак, которые меняют статус покемона (наследники **StatusMove**), скорее всего придется разобраться с классом **Effect**. Он позволяет на один или несколько ходов изменить состояние покемона или модификатор его базовых характеристик.
7. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

<p><b>Virizion</b></p>  <p><b>Атаки:</b></p> <ul style="list-style-type: none"><li>✓ Aerial Ace</li><li>✓ Calm Mind</li><li>✓ Giga Drain</li><li>✓ Leer</li></ul>	<p><b>Scyther</b></p>  <p><b>Атаки:</b></p> <ul style="list-style-type: none"><li>✓ Rest</li><li>✓ X-Scissor</li><li>✓ Leer</li></ul>	<p><b>Scizor</b></p>  <p><b>Атаки:</b></p> <ul style="list-style-type: none"><li>✓ Rest</li><li>✓ X-Scissor</li><li>✓ Leer</li><li>✓ Venoshock</li></ul>	<p><b>Happiny</b></p>  <p><b>Атаки:</b></p> <ul style="list-style-type: none"><li>✓ Fire Blast</li><li>✓ Psychic</li></ul>
<p><b>Chansey</b></p>  <p><b>Атаки:</b></p> <ul style="list-style-type: none"><li>✓ Fire Blast</li><li>✓ Psychic</li><li>✓ Mud Bomb</li></ul>	<p><b>Blissey</b></p>  <p><b>Атаки:</b></p> <ul style="list-style-type: none"><li>✓ Fire Blast</li><li>✓ Psychic</li><li>✓ Mud Bomb</li><li>✓ Minimize</li></ul>		

## Исходный код программы

1 contributor

27 lines (24 sloc) | 648 Bytes

RawBlame

```
1 package org.itmo.lab2;
2 import ru.ifmo.se.pokemon.*;
3 class BattlePok{
4     public static void main(String[] args){
5         Battle b = new Battle();
6         Pokemon p1 = new Pokemon("Чужой",1);
7         Pokemon p2 = new Pokemon("Хищник",1);
8         Pokemon p3 = new Virision("Халк",2);
9         Pokemon p4 = new Scyther("Дикарь",2);
10        Pokemon p5 = new Scizor("Силач",1);
11        Pokemon p6 = new Happiny("Цветочек",2);
12        Pokemon p7 = new Blissey("Гуру",1);
13        Pokemon p8 = new Chansey("Ветерок",2);
14        b.addAlly(p1);
15        b.addFoe(p2);
16        b.addAlly(p3);
17        b.addFoe(p4);
18        b.addAlly(p5);
19        b.addFoe(p6);
20        b.addFoe(p7);
21        b.addAlly(p8);
22        b.go();
23
24
25
26    }
27 }
```

MyProjects / src / org / itmo / lab2 / Blissey.java / <> Jump to ▾

Anastasyy Initial commit

History

1 contributor

11 lines (10 sloc) | 287 Bytes

...

```
1 package org.itmo.lab2;
2 import ru.ifmo.se.pokemon.*;
3 public class Blissey extends Pokemon{
4     public Blissey(String name, int level){
5         super(name,level);
6         setType(Type.PSYCHIC,Type.BUG);
7         setStats(1,1,6,4,6,3);
8         setMove(new FireBlast(),new Psychic(),new MudBomb(),new Minimize());
9     }
10
11 }
```

Закладки

Список для чтения

Закладки

Список для чтения

1 contributor

11 lines (10 sloc) | 273 Bytes

```
1 package org.itmo.lab2;
2 import ru.ifmo.se.pokemon.*;
3 public class Chansey extends Pokemon{
4     public Chansey(String name, int level){
5         super(name, level);
6         setType(Type.DRAGON, Type.STEEL);
7         setStats(2,3,6,2,6,3);
8         setMove(new FireBlast(), new Psychic(), new MudBomb());
9     }
10
11 }
```

Terms

Privacy

Security

Status

Docs

Contact GitHub

Pricing

API

Training

Blog

About

Закладки

Список для чтения

MyProjects / src / org / itmo / lab2 / Happiny.java / <> Jump to ▾


Anastasyy Initial commit 


History


1 contributor


11 lines (10 sloc) | 257 Bytes

```
1 package org.itmo.lab2;
2 import ru.ifmo.se.pokemon.*;
3 public class Happiny extends Pokemon{
4     public Happiny(String name, int level){
5         super(name, level);
6         setType(Type.ROCK, Type.WATER);
7         setStats(1,3,6,5,6,3);
8         setMove(new FireBlast(), new Psychic());
9     }
10
11 }
```

 Закладки

 Список для чтения

 Anastasy Initial commit

 History

1 contributor

10 lines (10 sloc) | 280 Bytes

1 package org.itmo.lab2;

2 import ru.ifmo.se.pokemon.\*;

3 public class Scizor extends Pokemon{

4 public Scizor(String name, int level){

5 super(name,level);


6 setType(Type.GRASS,Type.POISON);


7 setStats(1,2,2,5,6,3);


8 setMove(new Rest(),new XScissor(),new Leer(),new Venoshock ());


9 }

10 }

 Закладки

 Список для чтения

 Anastasy Initial commit

 History

1 contributor

12 lines (10 sloc) | 264 Bytes

1 package org.itmo.lab2;

2 import ru.ifmo.se.pokemon.\*;

3 public class Scyther extends Pokemon{

4 public Scyther(String name, int level){

5 super(name,level);

6 setType(Type.ICE,Type.DARK);

7 setStats(2,2,3,5,4,5);

8 setMove(new Rest(),new XScissor(),new Leer());

9 }

10 }

11



Anastasy Initial commit

History

1 contributor

15 lines (10 sloc) | 318 Bytes

...

```
1 package org.itmo.lab2;
2 import ru.ifmo.se.pokemon.*;
3 public class Virision extends Pokemon {
4     public Virision(String name, int level){
5         super(name, level);
6         setType(Type.FIRE, Type.FLYING);
7         setStats(3, 4, 5, 4, 5, 6);
8         setMove(new AerialAce(), new CalmMind(), new GigaDrain(), new Leer());
9     }
10 }
11 }
```



Закладки



Список для чтения

156 lines (128 sloc) | 3.13 KB

Raw

E

```
1 package org.itmo.lab2;
2 import ru.ifmo.se.pokemon.*;
3 class AerialAce extends SpecialMove{
4     public AerialAce(){
5         super(Type.GRASS, 60, 99999);
6     }
7
8     @Override
9     protected java.lang.String describe(){
10         return "Мочит";
11     }
12
13     @Override
14     protected void applyOppDamage(Pokemon def, double damage){
15         Status PokCon = def.getCondition();
16         if (PokCon.equals(Status.BURN) || PokCon.equals(Status.POISON) || PokCon.equals(Status.PARALYZE)) {
17             def.setMod(Stat.HP, (int) Math.round(damage) * 2);
18             Effect e = new Effect().chance(0.5).turns(3).stat(Stat.ATTACK, 1);
19             def.addEffect(e);
20         }
21     }
22 }
23
24
25 class CalmMind extends SpecialMove{
26     public CalmMind(){
27         super(Type.ICE, 15, 567);
28     }
29
30     @Override
31     protected java.lang.String describe(){
32         return "Мочит";
33     }
34 }
35
36
37
38 class GigaDrain extends SpecialMove{
39     public GigaDrain(){
```

Закладки

Список для чтения

Отправлено Вам

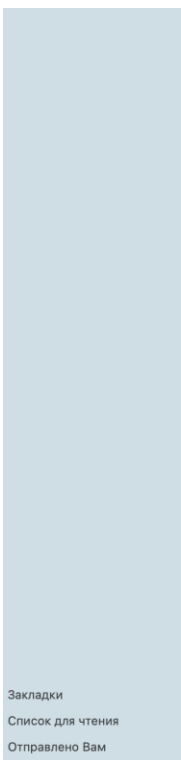
Закладки  
Список для чтения  
Отправлено Вам

```
34  
35 }  
36  
37  
38 class GigaDrain extends SpecialMove{  
39     public GigaDrain(){  
40         super(Type.DRAGON,5,1000);  
41     }  
42  
43     @Override  
44     protected java.lang.String describe(){  
45         return "Мочит";  
46     }  
47  
48 }  
49  
50 class Leer extends SpecialMove{  
51     public Leer(){  
52         super(Type.GHOST,902,10);  
53     }  
54  
55     @Override  
56     protected java.lang.String describe(){  
57         return "Мочит";  
58     }  
59     @Override  
60     protected void applyOppDamage(Pokemon def, double damage){  
61         Status PokCon = def.getCondition();  
62         if (PokCon.equals(Status.SLEEP) || PokCon.equals(Status.FREEZE) || PokCon.equals(Status.NORMAL)) {  
63             def.setMod(Stat.HP, (int) Math.round(damage) * 2);  
64             Effect e = new Effect().chance(0.5).turns(3).stat(Stat.ATTACK, 4);  
65             def.addEffect(e);  
66         }  
67     }  
68 }  
69  
70 class Rest extends SpecialMove{  
71     public Rest(){  
72         super(Type.FAIRY,45,10);  
73     }  
74  
75     @Override  
76     protected java.lang.String describe(){  
77         return "Летает";  
78     }  
79  
80 }  
81 class XScissor extends SpecialMove{  
82     public XScissor(){  
83         super(Type.FIRE,5,10);  
84     }  
85  
86     @Override  
87     protected java.lang.String describe(){  
88         return "Летает";  
89     }  
90 }  
91  
92 class Venoshock extends SpecialMove{  
93     public Venoshock(){  
94         super(Type.ICE,53,102);  
95     }  
96  
97     @Override  
98     protected java.lang.String describe(){  
99         return "Летает";  
100 }  
101  
102 }  
103 class FireBlast extends PhysicalMove{  
104     public FireBlast(){  
105         super(Type.GHOST,65,1021);  
106     }  
107  
108     @Override  
109     protected java.lang.String describe(){  
110         return "Пулсаetre";  
111     }  
112 }
```

Закладки  
Список для чтения  
Отправлено Вам

```
70 class Rest extends SpecialMove{  
71     public Rest(){  
72         super(Type.FAIRY,45,10);  
73     }  
74  
75     @Override  
76     protected java.lang.String describe(){  
77         return "Летает";  
78     }  
79  
80 }  
81 class XScissor extends SpecialMove{  
82     public XScissor(){  
83         super(Type.FIRE,5,10);  
84     }  
85  
86     @Override  
87     protected java.lang.String describe(){  
88         return "Летает";  
89     }  
90 }  
91  
92 class Venoshock extends SpecialMove{  
93     public Venoshock(){  
94         super(Type.ICE,53,102);  
95     }  
96  
97     @Override  
98     protected java.lang.String describe(){  
99         return "Летает";  
100 }  
101  
102 }  
103 class FireBlast extends PhysicalMove{  
104     public FireBlast(){  
105         super(Type.GHOST,65,1021);  
106     }  
107  
108     @Override  
109     protected java.lang.String describe(){  
110         return "Пулсаetre";  
111     }  
112 }
```





```
114 class Psychic extends PhysicalMove{
115     public Psychic(){
116         super(Type.GRASS,513,1842);
117     }
118
119     @Override
120     protected java.lang.String describe(){
121         return "Плывается";
122     }
123 }
124
125 class MudBomb extends PhysicalMove{
126     public MudBomb(){
127         super(Type.BUG,33,142);
128     }
129
130     @Override
131     protected java.lang.String describe(){
132         return "Взрывается";
133     }
134 }
135
136 class Minimize extends PhysicalMove{
137     public Minimize(){
138         super(Type.NONE,312,1421);
139     }
140
141     @Override
142     protected java.lang.String describe(){
143         return "Взрывается";
144     }
145
146     @Override
147     protected void applyOppDamage(Pokemon def, double damage){
148         Status PokCon = def.getCondition();
149         if (PokCon.equals(Status.FREEZE) || PokCon.equals(Status.POISON) || PokCon.equals(Status.SLEEP)) {
150             def.setMod(Stat.HP, (int) Math.round(damage) * 2);
151             Effect e = new Effect().chance(0.5).turns(3).stat(Stat.ATTACK, -4);
152             def.addEffect(e);
153         }
154     }
155 }
```

## Ссылка

<https://github.com/AnastasyMyProjects/tree/main/src/org/itmo/lab2>

### Компиляция и разворачивание программы

javac -classpath ./classes:/lib/Pokemon.jar -d ./classes src/org/itmo/lab2/NewAttakcsPok.java  
javac -cp ./classes:/lib/Pokemon.jar -d ./classes src/org/itmo/lab2/Virision.java (другие классы)

javac -cp ./classes:/lib/Pokemon.jar -d ./classes src/org/itmo/lab2/BattlePok.java

java -classpath ./classes:/lib/Pokemon.jar org.itmo.lab2.BattlePok

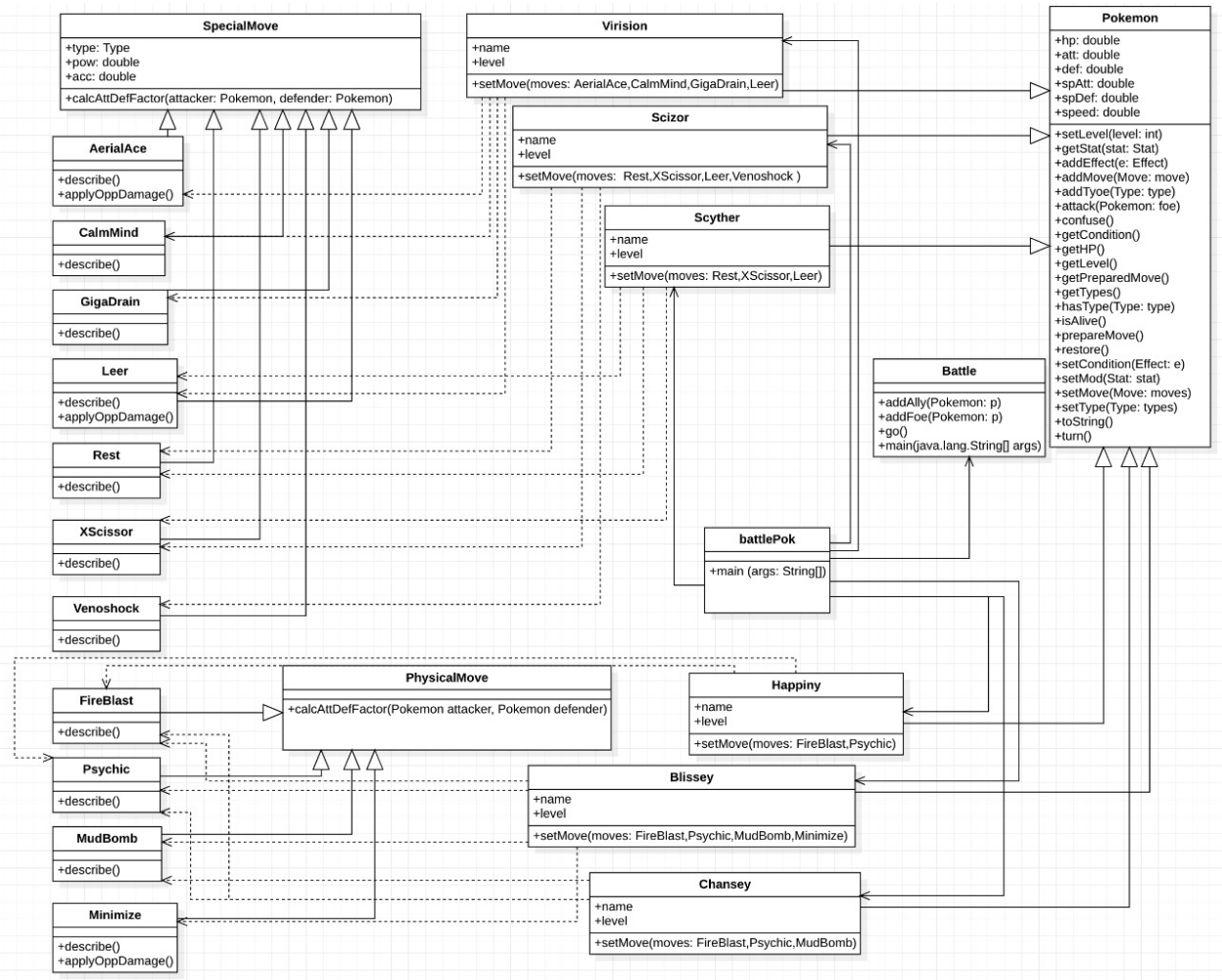
создание manifest.mf

jar cvmf manifest.mf BattlePok.jar -C ./classes org/itmo/lab2

java -jar BattlePok.jar

jar tvf battlePok.jar

## Диаграмма классов



## Выводы

Познакомилась с принципами и элементами ООП такие как инкапсуляция, наследование и полиморфизм. Разработала собственные классы и конструкторы, научилась разрабатывать методы.

