

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования

«Национальный исследовательский университет ИТМО»  
Факультет программной инженерии и компьютерной техники

Лабораторная работа №2

Вариант 32710

Группа: Р3132

Выполнил: Овчаренко Александр Андреевич

Санкт-Петербург

2021

## 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.



## Torracat



### Атаки:

- ✂ Facade
- ✂ Swagger
- ✂ Mud-Slap

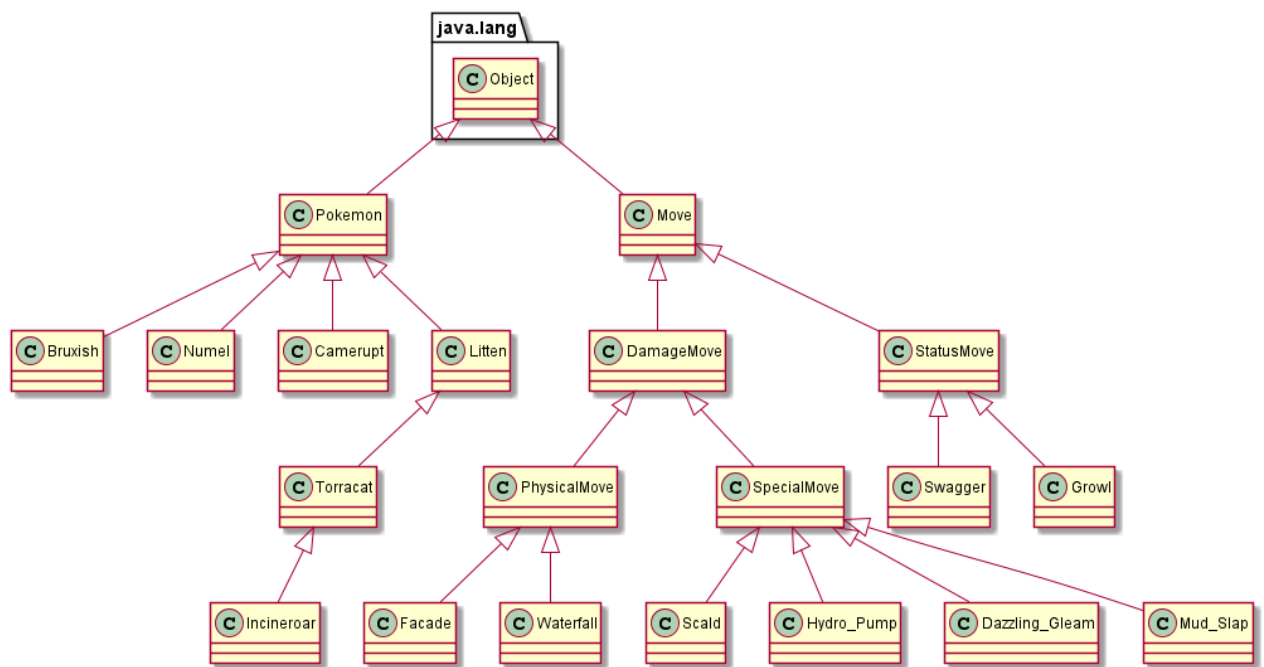
## Incineroar



### Атаки:

- ✂ Facade
- ✂ Swagger
- ✂ Mud-Slap
- ✂ Growl

## Выполнение



```
1  import ru.ifmo.se.pokemon.*;
2
3  class Main {
4      public static void main(String[] args) {
5          Battle b = new Battle();
6          Bruxish p1 = new Bruxish();
7          Numel p2 = new Numel();
8          Camerupt p3 = new Camerupt();
9          Litten p4 = new Litten();
10         Torracat p5 = new Torracat();
11         Incineroar p6 = new Incineroar();
12         b.addAlly(p1);
13         b.addFoe(p2);
14         b.addAlly(p3);
15         b.addFoe(p4);
16         b.addAlly(p5);
17         b.addFoe(p6);
18         b.go();
19     }
20 }
21
22 class Bruxish extends Pokemon{
23
24     Bruxish() {
25         super("Bruxish", 1);
26         setType(Type.WATER, Type.PSYCHIC);
27         setStats(68.0D, 105.0D, 70.0D, 70.0D, 70.0D, 92.0D);
28         setMove(new Facade(), new Waterfall(), new Scald(), new Swagger());
29     }
30 }
```

```

31     Bruxish(int lv) {
32         super("Bruxish", lv);
33         setType(Type.WATER, Type.PSYCHIC);
34         setStats(68.0D, 105.0D, 70.0D, 70.0D, 70.0D, 92.0D);
35         setMove(new Facade(), new Waterfall(), new Scald(), new Swagger());
36     }
37
38     Bruxish(String nm, int lv) {
39         super("Bruxish", 1);
40         setType(Type.WATER, Type.PSYCHIC);
41         setStats(68.0D, 105.0D, 70.0D, 70.0D, 70.0D, 92.0D);
42         setMove(new Facade(), new Waterfall(), new Scald(), new Swagger());
43     }
44 }
45
46 class Numel extends Pokemon{
47
48     Numel() {
49         super("Numel", 1);
50         setType(Type.FIRE, Type.GROUND);
51         setStats(60.0D, 60.0D, 40.0D, 65.0D, 45.0D, 35.0D);
52         setMove(new Scald(), new Hydro_Pump(), new Facade());
53     }
54
55     Numel(int lv) {
56         super("Numel", lv);
57         setType(Type.FIRE, Type.GROUND);
58         setStats(60.0D, 60.0D, 40.0D, 65.0D, 45.0D, 35.0D);
59         setMove(new Scald(), new Hydro_Pump(), new Facade());
60     }
61 }

```

```

62     Numel(String nm, int lv) {
63         super(nm, lv);
64         setType(Type.FIRE, Type.GROUND);
65         setStats(60.0D, 60.0D, 40.0D, 65.0D, 45.0D, 35.0D);
66         setMove(new Scald(), new Hydro_Pump(), new Facade());
67     }
68 }
69
70 class Camerupt extends Numel{
71
72     Camerupt() {
73         super("Camerupt", 1);
74         setType(Type.FIRE, Type.GROUND);
75         setStats(70.0D, 100.0D, 70.0D, 105.0D, 75.0D, 40.0D);
76         setMove(new Scald(), new Hydro_Pump(), new Facade(), new Dazzling_Gleam());
77     }
78
79     Camerupt(int lv) {
80         super("Camerupt", lv);
81         setType(Type.FIRE, Type.GROUND);
82         setStats(70.0D, 100.0D, 70.0D, 105.0D, 75.0D, 40.0D);
83         setMove(new Scald(), new Hydro_Pump(), new Facade(), new Dazzling_Gleam());
84     }
85
86     Camerupt(String nm, int lv) {
87         super(nm, lv);
88         setType(Type.FIRE, Type.GROUND);
89         setStats(70.0D, 100.0D, 70.0D, 105.0D, 75.0D, 40.0D);
90         setMove(new Scald(), new Hydro_Pump(), new Facade(), new Dazzling_Gleam());
91     }
92 }

```

```

94  class Litten extends Pokemon{
95
96      Litten() {
97          super("Litten", 1);
98          setType(Type.FIRE);
99          setStats(45.0D, 65.0D, 40.0D, 60.0D, 40.0D, 70.0D);
100         setMove(new Facade(), new Swagger());
101     }
102
103     Litten(int lv) {
104         super("Litten", lv);
105         setType(Type.FIRE);
106         setStats(45.0D, 65.0D, 40.0D, 60.0D, 40.0D, 70.0D);
107         setMove(new Facade(), new Swagger());
108     }
109
110     Litten(String nm, int lv) {
111         super(nm, lv);
112         setType(Type.FIRE);
113         setStats(45.0D, 65.0D, 40.0D, 60.0D, 40.0D, 70.0D);
114         setMove(new Facade(), new Swagger());
115     }
116 }
117
118 class Torracat extends Litten{
119
120     Torracat() {
121         super("Torracat", 1);
122         setType(Type.FIRE);
123         setStats(65.0D, 85.0D, 50.0D, 80.0D, 50.0D, 90.0D);
124         setMove(new Facade(), new Swagger(), new Mud_Slap());
125     }

```

```

127     Torracat(int lv) {
128         super("Torracat", lv);
129         setType(Type.FIRE);
130         setStats(65.0D, 85.0D, 50.0D, 80.0D, 50.0D, 90.0D);
131         setMove(new Facade(), new Swagger(), new Mud_Slap());
132     }
133
134     Torracat(String nm, int lv) {
135         super(nm, lv);
136         setType(Type.FIRE);
137         setStats(65.0D, 85.0D, 50.0D, 80.0D, 50.0D, 90.0D);
138         setMove(new Facade(), new Swagger(), new Mud_Slap());
139     }
140 }
141
142 class Incineroar extends Torracat{
143
144     Incineroar() {
145         super("Incineroar", 1);
146         setType(Type.FIRE, Type.DARK);
147         setStats(95.0D, 115.0D, 90.0D, 80.0D, 90.0D, 90.0D);
148         setMove(new Facade(), new Swagger(), new Mud_Slap(), new Growl());
149     }
150
151     Incineroar(int lv) {
152         super("Incineroar", lv);
153         setType(Type.FIRE, Type.DARK);
154         setStats(95.0D, 115.0D, 90.0D, 80.0D, 90.0D, 90.0D);
155         setMove(new Facade(), new Swagger(), new Mud_Slap(), new Growl());
156     }
157 }

```

```

158 Incineroar(String nm, int lv) {
159     super(nm, lv);
160     setType(Type.FIRE, Type.DARK);
161     setStats(95.0D, 115.0D, 90.0D, 80.0D, 90.0D, 90.0D);
162     setMove(new Facade(), new Swagger(), new Mud_Slap(), new Growl());
163 }
164
165 s Facade extends PhysicalMove {
166     Facade() {super(Type.NORMAL, 70.0D, 100.0D);}
167
168     protected double calcBaseDamage(Pokemon att, Pokemon def) {
169         if (def.getCondition() == Status.BURN | def.getCondition() == Status.POISON |
170             def.getCondition() == Status.PARALYZE) return (0.4D * (double)att.getLevel() + 2.0D) *
171             this.power * 2 / 150.0D;
172         else return (0.4D * (double)att.getLevel() + 2.0D) * this.power / 150.0D;
173     }
174     protected String describe() {return "использует Facade";}
175 }
176
177 s Waterfall extends PhysicalMove {
178     Waterfall() {super(Type.WATER, 80.0D, 100.0D);}
179 }
180 protected String describe() {return "использует Waterfall";}
181 }
182 }
183
184 s Scald extends SpecialMove {
185     Scald() {super(Type.WATER, 80.0D, 100.0D);}
186 }
187 protected void applyOppEffects(Pokemon p) {
188     int z = 0;

```

```

188         int z = 0;
189         if (Math.random() <= 0.3D) {
190             for (Type t: p.getTypes()) {
191                 if (t == Type.FIRE) {
192                     z = -1;
193                 }
194             }
195             if (z == 0) Effect.burn(p);
196         }
197     }
198     protected String describe() {return "использует Scald";}
199 }
200 }
201
202 s Swagger extends StatusMove {
203     Swagger() {super(Type.NORMAL, 0.0D, 85.0D);}
204 }
205 protected void applyOppEffects(Pokemon p) {
206     p.confuse();
207     p.setMod(Stat.ATTACK, +2);
208 }
209 protected String describe() {return "использует Swagger";}
210 }
211 }
212
213 s Hydro_Pump extends SpecialMove {
214     Hydro_Pump() {super(Type.WATER, 110.0D, 80.0D);}
215     protected String describe() {return "использует Hydro_Pump";}
216 }
217 }
218

```

```

219 s Dazzling_Gleam extends SpecialMove {
220 Dazzling_Gleam() {super(Type.FAIRY, 80.0D, 100.0D);}
221 protected String describe() {return "использует Dazzling_Gleam";
222 }
223 }
224
225 s Mud_Slap extends SpecialMove {
226 Mud_Slap() {super(Type.GROUND, 20.0D, 100.0D);}
227
228 protected void applyOppEffects(Pokemon p) {
229     p.setMod(Stat.ACCURACY, -1);
230 }
231 protected String describe() {return "использует Mud_Slap";
232 }
233 }
234
235 s Growl extends StatusMove {
236 Growl() {super(Type.NORMAL, 0.0D, 100.0D);}
237
238 protected void applyOppEffects(Pokemon p) {
239     p.setMod(Stat.ATTACK, -1);
240 }
241 protected String describe() {return "использует Growl";}
242
243

```

Вывод программы при исполнении кода пункта 3:

```

' '@C:\Users\sasao\AppData\Local\Temp\cp_7ufp8mv6h8xqcwmt77fp83usb.argfile' 'Main'
Pokemon Alien из команды белых вступает в бой!
Pokemon Predator из команды полосатых вступает в бой!
Pokemon Alien борется с соперником.
Pokemon Predator теряет 4 здоровья.
Pokemon Alien теряет 1 здоровья.

Pokemon Predator борется с соперником.
Pokemon Alien теряет 5 здоровья.
Pokemon Predator теряет 1 здоровья.

Pokemon Predator борется с соперником.
Pokemon Predator теряет 3 здоровья.
Pokemon Predator теряет 1 здоровья.

Pokemon Predator борется с соперником.
Pokemon Predator теряет 3 здоровья.
Pokemon Predator теряет 1 здоровья.
Оба покемона теряют сознание.
В команде полосатых не осталось покемонов.
Команда белых побеждает в этом бою!
PS C:\Users\sasao\Documents\Java_ITMO\src\lab_2>

```



Вывод программы при исполнении кода на первом изображении:

```
Bruxish Bruxish из команды фиолетовых вступает в бой!  
Numel Numel из команды полосатых вступает в бой!  
Bruxish Bruxish использует Waterfall.  
Numel Numel теряет 28 здоровья.  
Numel Numel теряет сознание.  
Litten Litten из команды полосатых вступает в бой!  
Bruxish Bruxish использует Facade.  
Litten Litten теряет 6 здоровья.  
  
Litten Litten использует Facade.  
Bruxish Bruxish теряет 4 здоровья.  
  
Bruxish Bruxish использует Swagger.  
Litten Litten увеличивает атаку.  
  
Litten Litten использует Facade.  
Bruxish Bruxish теряет 4 здоровья.  
  
Bruxish Bruxish использует Facade.  
Litten Litten теряет 7 здоровья.  
Litten Litten теряет сознание.  
Incineroar Incineroar из команды полосатых вступает в бой!  
Bruxish Bruxish использует Waterfall.  
Incineroar Incineroar теряет 18 здоровья.  
Incineroar Incineroar теряет сознание.  
В команде полосатых не осталось покемонов.  
Команда фиолетовых побеждает в этом бою!  
PS C:\Users\sasao\Documents\Java_ITMO\src\lab_2>
```

## **Итог**

Я научился подключать jar-архив к своему проекту. Я разобрался как устроено объектно-ориентированное программирование, узнал об основных понятиях ООП. Я изучил, какие есть модификаторы доступа, как переопределять методы, как создавать конструкторы, что такое перегрузка методов и конструкторов.