

Национальный исследовательский Университет ИТМО
Мегафакультет компьютерных технологий и управления
Факультет программной инженерии и компьютерной техники

Программирование

Лабораторная работа №2

Работу

выполнил:

Д. С. Зубахин

Группа: Р3131

Преподаватель:

И. В. Усков

Санкт-Петербург
2021

Содержание

1. Текст задания	3
2. Диаграмма классов реализованной объектной модели	4
3. Исходный код программы	5
3.1. Main	5
3.2. Атаки	5
3.3. Покемоны	9
4. Результат работы программы	12
5. Выводы по работе	14

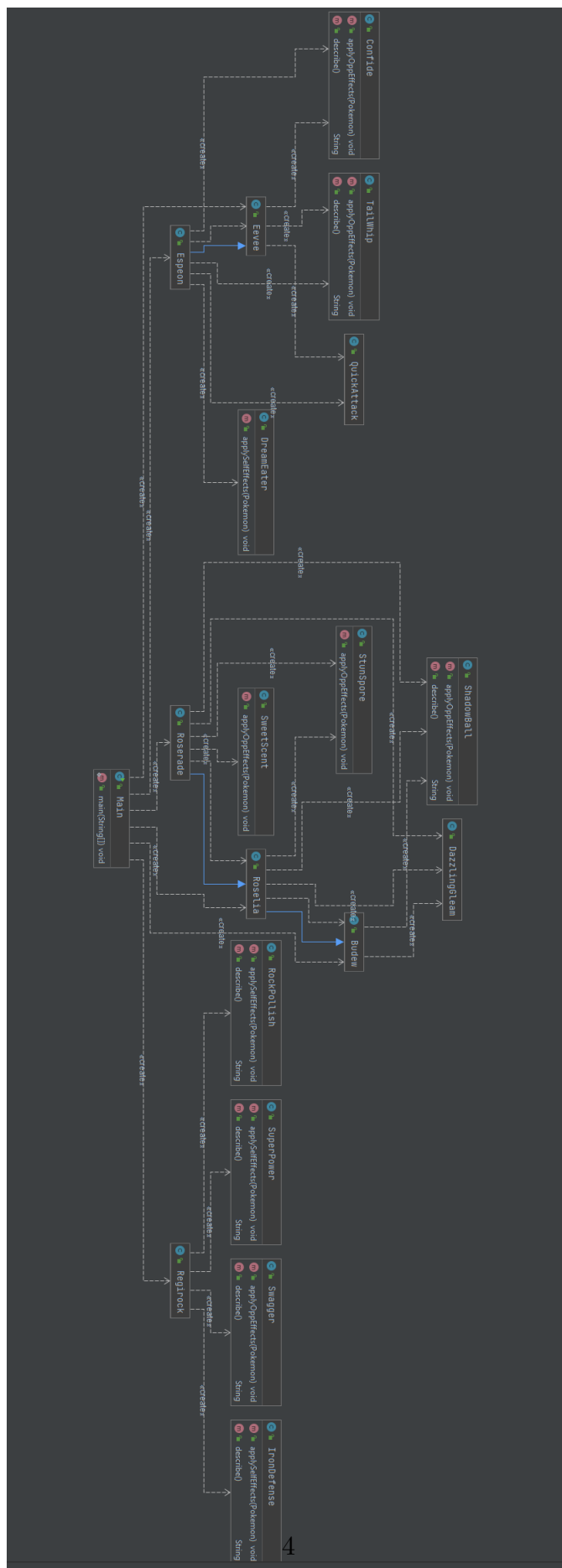
1. Текст задания

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.

```
1 Battle b = new Battle();
2 Pokemon p1 = new Pokemon("Чужой", 1);
3 Pokemon p2 = new Pokemon("Хищник", 1);
4 b.addAlly(p1);
5 b.addFoe(p2);
6 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.

2. Диаграмма классов реализованной объектной модели



3. Исходный код программы

3.1. Main

```
1  import ru.ifmo.se.pokemon.*;
2
3  public class Main {
4
5      public static void main(String[] args) {
6          Battle dom = new Battle();
7          dom.addAlly(new Regirock("Игорь", 1));
8          dom.addAlly(new Eevee("Аркадий", 1));
9          dom.addAlly(new Espeon("Паша", 1));
10         dom.addFoe(new Budew("Дима", 1));
11         dom.addFoe(new Roselia("Валера", 1));
12         dom.addFoe(new Roserade("Вазген", 1));
13         dom.go();
14     }
15 }
```

3.2. Атаки

Confide

```
1  package PockemonMove;
2  import ru.ifmo.se.pokemon.*;
3
4  public class Confide extends StatusMove{
5      public Confide(){
6          super(Type.NORMAL, 0,100);
7      }
8      @Override
9      public void applyOppEffects(Pokemon p) {
10         p.setMod(Stat.ATTACK, -1);
11     }
12     @Override
13     public String describe(){
14         return "Уменьшает атаку противника на 1 ступень";
15     }
16 }
```

DazzlingGleam

```
1  package PockemonMove;
2  import ru.ifmo.se.pokemon.*;
3
4  public class DazzlingGleam extends PhysicalMove{
5      public DazzlingGleam() {
6          super(Type.FAIRY, 55,100);
7      }
8  }
```

DreamEater

```
1 package PockemonMove;
2 import ru.ifmo.se.pokemon.*;
3
4 public class DreamEater extends SpecialMove{
5     public DreamEater() {
6         super(Type.PSYCHIC,100,100);
7     }
8     @Override
9     public void applySelfEffects(Pokemon p){
10         p.setMod(Stat.HP, 5);
11     }
12 }
```

IronDefense

```
1 package PockemonMove;
2 import ru.ifmo.se.pokemon.*;
3
4 public class IronDefense extends StatusMove{
5     public IronDefense() {
6         super(Type.STEEL,0,100);
7     }
8     @Override
9     public void applySelfEffects(Pokemon p){
10         p.setMod(Stat.DEFENSE, 2);//
11         ↪ "повышает Защиту пользователя на две ступени"
12     }
13     @Override
14     public String describe() {
15         return "увеличил свою защиту на 2 ступени";
16     }
17 }
```

QuickAttack

```
1 package PockemonMove;
2 import ru.ifmo.se.pokemon.*;
3
4 public class QuickAttack extends PhysicalMove{
5     public QuickAttack() {
6         super(Type.NORMAL, 40,100);
7     }
8 }
```

RockPolish

```
1 package PockemonMove;
2 import ru.ifmo.se.pokemon.*;
3
4 public class RockPolish extends StatusMove{
5     public RockPolish() {
6         super(Type.GROUND, 0, 100);
7     }
8     @Override
9     public void applySelfEffects(Pokemon p) {
10         p.setMod(Stat.SPEED, 2); // повышает скорость пользователя на две
11         ↪ ступени.
12     }
13     @Override
14     public String describe(){
15         return "увеличил свою скорость на два порядка";
16     }
17 }
```

ShadowBall

```
1 package PockemonMove;
2 import ru.ifmo.se.pokemon.*;
3
4 public class ShadowBall extends SpecialMove{
5     public ShadowBall() {
6         super(Type.GHOST, 80, 100);
7     }
8     @Override
9     public void applyOppEffects(Pokemon p) {
10         p.setMod(Stat.DEFENSE, 4/5);
11         p.getHP();
12     }
13     @Override
14     public String describe() {
15         return "снизил атаку противника на 20%";
16     }
17 }
```

StunSpore

```
1 package PockemonMove;
2 import ru.ifmo.se.pokemon.*;
3
4 public class StunSpore extends SpecialMove{
5     public StunSpore() {
6         super(Type.GRASS, 0, 75);
7     }
8     @Override
9     public void applyOppEffects(Pokemon p){
10         Effect.paralyze(p);
11     }
12 }
```

SuperPower

```
1 package PockemonMove;
2 import ru.ifmo.se.pokemon.*;
3
4
5 public class SuperPower extends PhysicalMove{
6     public SuperPower() {
7         super(Type.FIGHTING, 120, 100);
8     }
9     @Override
10    public void applySelfEffects(Pokemon p) {
11        p.setMod(Stat.ATTACK, -1);
12        p.setMod(Stat.DEFENSE, -1);
13    }
14    @Override
15    public String describe() {
16        return "снизил свою атаку и защиту на 1 порядок";
17    }
18 }
```

Swagger

```
1 package PockemonMove;
2 import ru.ifmo.se.pokemon.*;
3
4 public class Swagger extends StatusMove{
5     public Swagger() {
6         super(Type.NORMAL, 0, 100);
7     }
8     @Override
9     public void applyOppEffects(Pokemon p){
10        p.setMod(Stat.ATTACK,
11        ↪ 2); // "повышает атаку противника на две ступени"
12    }
13    @Override
14    public String describe(){
15        return "повысил атаку противника на 2 порядка" ;
16    }
17 }
```

SweetScent

```
1 package PockemonMove;
2 import ru.ifmo.se.pokemon.*;
3
4 public class SweetScent extends StatusMove{
5     public SweetScent() {
6         super(Type.NORMAL, 0, 100);
7     }
8     @Override
9     public void applyOppEffects(Pokemon p){
10        p.setMod(Stat.EVASION, -5);
11    }
12 }
```



```
12 }
```

TailWhip

```
1 package PockemonMove;
2 import ru.ifmo.se.pokemon.*;
3
4 public class TailWhip extends StatusMove{
5     public TailWhip() {
6         super(Type.NORMAL, 0,100);
7     }
8     @Override
9     public void applyOppEffects(Pokemon p) {
10         p.setMod(Stat.DEFENSE, -1);
11     }
12     @Override
13     public String describe(){
14         return "уменьшает защиту противника на 1 ступень";
15     }
16 }
```

3.3. Покемоны

Budew

```
1 import ru.ifmo.se.pokemon.*;
2 import PockemonMove.*;
3
4 public class Budew extends Pokemon{
5     public Budew(String name, int level){
6         super(name,level);
7         setStats(40,30,35,50,70,55);//характеристики
8         setType(Type.GRASS);//тип
9         setMove(new ShadowBall(), new DazzlingGleam());//ставим атаки
10    }
11 }
```

Eevee

```
1 import ru.ifmo.se.pokemon.*;
2 import PockemonMove.*;
3
4
5 public class Eevee extends Pokemon{
6     public Eevee(String name, int level){
7         super(name,level);
8         setStats(55,55,50,45,65,55);
9         setType(Type.NORMAL);
10        setMove(new QuickAttack(), new TailWhip(), new Confide());
11    }
12 }
```

Espeon

```
1  import ru.ifmo.se.pokemon.*;
2  import PockemonMove.*;
3
4  public class Espeon extends Pokemon{
5      public Espeon(String name, int level){
6          super(name, level);
7          setStats(65,65, 60, 130,95,110);//
8          setType(Type.PSYCHIC);
9          setMove(new QuickAttack(), new TailWhip(), new Confide(), new
10             ↵ DreamEater());
11     }
}
```

Regirock

```
1  import ru.ifmo.se.pokemon.*;
2  import PockemonMove.*;
3
4  public class Regirock extends Pokemon {
5      public Regirock (String name, int level){
6          super(name, level);
7          setStats(80, 100, 200, 50, 100, 50);
8          setType(Type.GROUND);
9          setMove(new SuperPower(), new IronDefense(), new Swagger(), new
10             ↵ RockPollish());
11     }
}
```

Roselia

```
1  import ru.ifmo.se.pokemon.*;
2  import PockemonMove.*;
3
4  public class Roselia extends Pokemon {
5      public Roselia(String name, int level){
6          super(name, level);
7          setStats(50,60,45,100,80,65);
8          setType(Type.GRASS);
9          setMove(new ShadowBall(), new DazzlingGleam(), new StunSpore());
10     }
11 }
```

Roserade

```
1  import ru.ifmo.se.pokemon.*;
2  import PockemonMove.*;
3
4  public class Roserade extends Pokemon{
5      public Roserade (String name, int level){
6          super(name, level);
7          setStats(60,70,65,125,105,90);
8          setType(Type.GRASS);
9          setMove( new ShadowBall(), new DazzlingGleam(), new StunSpore(),
10                  ↵ new SweetScent());
11      }
```

4. Результат работы программы

```
1 Regirock Игорь из команды полосатых вступает в бой!
2 Budew Дима из команды фиолетовых вступает в бой!
3 Budew Дима атакует.
4 Regirock Игорь теряет 5 здоровья.
5
6 Regirock Игорь увеличил свою защиту на 2 ступени.
7 Regirock Игорь увеличивает защиту.
8
9 Budew Дима атакует.
10 Regirock Игорь теряет 3 здоровья.
11
12 Regirock Игорь увеличил свою скорость на два порядка.
13 Regirock Игорь увеличивает скорость.
14
15 Regirock Игорь снизил свою атаку и защиту на 1 порядок.
16 Budew Дима теряет 7 здоровья.
17 Regirock Игорь уменьшает атаку.
18 Regirock Игорь уменьшает защиту.
19
20 Budew Дима атакует.
21 Regirock Игорь теряет 3 здоровья.
22
23 Regirock Игорь повысил атаку противника на 2 порядка.
24 Budew Дима увеличивает атаку.
25
26 Budew Дима снизил атаку противника на 20%.
27 Regirock Игорь теряет 4 здоровья.
28 Regirock Игорь теряет сознание.
29 Eevee Аркадий из команды полосатых вступает в бой!
30 Budew Дима снизил атаку противника на 20%.
31 Eevee Аркадий теряет 1 здоровья.
32 Eevee Аркадий не замечает воздействие типа GHOST
33
34 Eevee Аркадий атакует.
35 Budew Дима теряет 5 здоровья.
36 Budew Дима теряет сознание.
37 Roselia Валера из команды фиолетовых вступает в бой!
38 Roselia Валера атакует.
39 Eevee Аркадий теряет 4 здоровья.
40
41 Eevee Аркадий атакует.
42 Roselia Валера теряет 5 здоровья.
43
44 Roselia Валера снизил атаку противника на 20%.
45 Eevee Аркадий теряет 1 здоровья.
46 Eevee Аркадий не замечает воздействие типа GHOST
47
48 Eevee Аркадий уменьшает защиту противника на 1 ступень.
49 Roselia Валера уменьшает защиту.
50
51 Roselia Валера атакует.
52 Eevee Аркадий теряет 4 здоровья.
53
54 Eevee Аркадий атакует.
55 Roselia Валера теряет 6 здоровья.
56
57 Roselia Валера атакует.
```

58 Eevee Аркадий теряет 5 здоровья.
59 Eevee Аркадий теряет сознание.
60 Espeon Паша из команды полосатых вступает в бой!
61 Espeon Паша атакует.
62 Roselia Валера теряет 6 здоровья.
63 Espeon Паша теряет 5 здоровья.
64 Roselia Валера теряет сознание.
65 Roserade Вазген из команды фиолетовых вступает в бой!
66 Espeon Паша Уменьшает атаку противника на 1 ступень.
67 Roserade Вазген уменьшает атаку.
68
69 Roserade Вазген атакует.
70 Espeon Паша теряет 4 здоровья.
71
72 Espeon Паша атакует.
73 Roserade Вазген теряет 11 здоровья.
74 Espeon Паша теряет 5 здоровья.
75 Espeon Паша теряет сознание.
76 В команде полосатых не осталось покемонов.
77 Команда фиолетовых побеждает в этом бою!

5. Выводы по работе

Мы разобрались в основных понятиях Объектно-ориентированного программирования, а именно в понятиях: объекты, наследование, полиморфизм, инкапсуляция. Мы научились создавать классы и экземпляры классов, а также расширять классы. Мы научились работать с модификаторами доступа и вызывать методы. Научились импортировать сторонние библиотеки в idea и работать с ними.