Санкт-Петербургский Национальный Исследовательский
Университет ИТМО
Факультет Программной Инженерии и Компьютерной Техники

Вариант №335129
Лабораторная работа №3
по дисциплине
"Программирование"

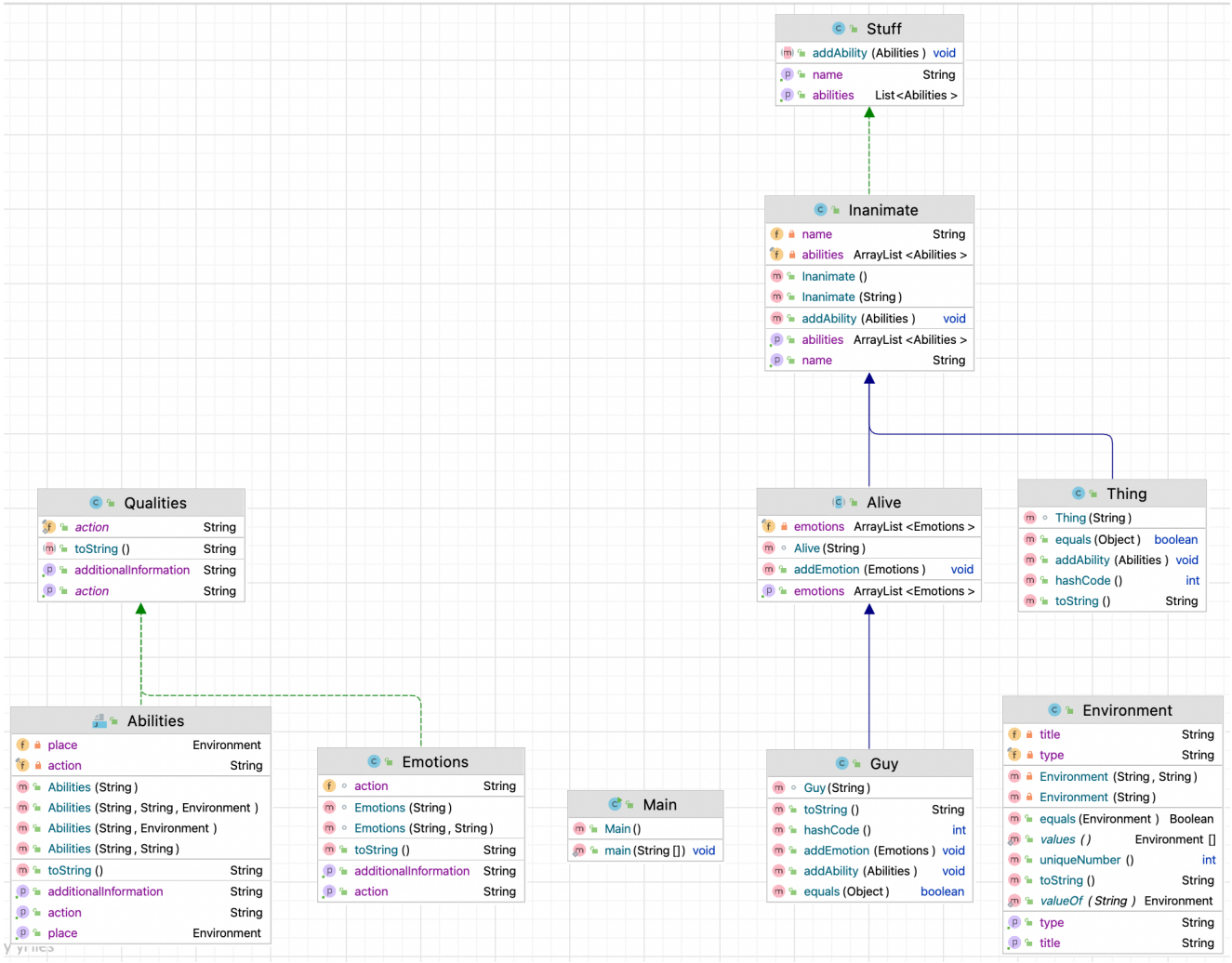
Выполнила студентка группы P3117
Галина Игнатова
Преподаватель:
Письмак Алексей Евгеньевич

г. Санкт-Петербург
2021г.

1 Текст задания

Незнайка и Козлик с завистью поглядывали на коротышек, которые сидели у ресторанов за столиками и угощались разными вкусными блюдами. Смотреть на все это и не иметь возможности утолить голод было очень мучительно. Они принялись бродить вдоль выставленных у краев тротуара кривых зеркал и разглядывать свои отражения. Одно из зеркал до такой степени исказило их физиономии, что Незнайка и Козлик, как ни было им грустно, все же не смогли удержаться от смеха.

2 Диаграмма классов реализованной объектной модели



3 Исходный код программы

Листинг 1: Main.java

```
1  public class Main {
2      public static void main(String[] args){
3          Guy noknow = new Guy("Незнайка");
4          noknow.addAbility(new Abilities("смотреть", "на␣коротышек", Environment.
               RESTAURANT));
```

```
        noknow.addEmotion(new Emotions("зависть"));

        Guy goat = new Guy("Козлик");
        goat.addAbility(new Abilities("смотреть", "на_коротышек", Environment.RESTAURANT)
            );
        goat.addEmotion(new Emotions("зависть"));

        Guy shorts = new Guy("Коротышки");
        shorts.addAbility(new Abilities("угощаться", "разными_вкусными_блюдами",
            Environment.RESTAURANT));

        noknow.addAbility(new Abilities("ходить", Environment.STREET));
        noknow.addAbility(new Abilities("бродить", Environment.STREET));

        Thing mirror = new Thing("Зеркало");
        mirror.addAbility(new Abilities("криво_отражать", "физиономии"));

        goat.addEmotion(new Emotions("радость"));
        noknow.addEmotion(new Emotions("счастье"));
    }
}
```

Листинг 2: Qualities.java

```
public interface Qualities {
    String action = "";
    String extra = "";

    String getAction();
    String getAdditionalInformation();
    String toString();
}
```

Листинг 3: Abilities.java

```
public class Abilities implements Qualities{
    private final String action;
    private Environment place = Environment.EVERYWHERE;
    private String extra = "";

    public Abilities(String ability) {
        this.action = ability;
    }

    public Abilities(String ability, String extra) {
        this(ability);
        this.extra = extra;
    }

    public Abilities(String ability, Environment place) {
        this(ability);
        this.place = place;
    }
```

```
19
20      public Abilities(String ability, String extra, Environment place) {
21          this(ability, extra);
22          this.place = place;
23      }
24
25      public String getAction(){
26          return action;
27      }
28
29      public Environment getPlace(){
30          return place;
31      }
32
33      public String getAdditionalInformation(){
34          return extra + "␣" + place;
35      }
36
37      public String toString(){
38          if (place.equals(Environment.EVERYWHERE)) {
39              return action + "␣" + extra;
40          } else {
41              return action + "␣" + extra + "␣" + place;
42          }
43      }
44  }
```

Листинг 4: Emotions.java

```
1   public class Emotions implements Qualities{
2       String action = "";
3       String extra = "";
4
5       Emotions(String action){
6           this.action = action;
7       }
8
9       Emotions(String action, String extra){
10          this(action);
11          this.extra = extra;
12      }
13
14      public String getAction(){
15          return action;
16      }
17
18      public String getAdditionalInformation(){
19          return extra;
20      }
21
22      public String toString(){
23          return action + "␣" + extra;
```

```
24      }
25  }
```

Листинг 5: Stuff.java

```
1  import java.util.List;
2
3  public interface Stuff {
4      String getName();
5      List<Abilities> getAbilities();
6      void addAbility(Abilities ability);
7  }
```

Листинг 6: Inanimate.java

```
1  import java.util.ArrayList;
2
3  public abstract class Inanimate implements Stuff {
4      private String name;
5      private final ArrayList<Abilities> abilities = new ArrayList<>();
6
7      public Inanimate(){
8      }
9
10     public Inanimate(String name){
11         this.name = name;
12     }
13
14     public String getName(){
15         return name;
16     }
17
18     public ArrayList<Abilities> getAbilities(){
19         return abilities;
20     }
21
22     public void addAbility(Abilities ability){
23         abilities.add(ability);
24     }
25 }
```

Листинг 7: Thing.java

```
1  public class Thing extends Inanimate{
2      Thing(String name){
3          super(name);
4          System.out.println("Объект " + name + " успешно создан.");
5      }
6
7      @Override
8      public void addAbility(Abilities ability) {
9          super.addAbility(ability);
10         System.out.println(getName() + " умеет " + ability + ".");
11     }
```

```
12
13      @Override
14      public int hashCode() {
15          return super.hashCode()+this.getName().hashCode();
16      }
17      @Override
18      public boolean equals(Object obj) {
19          return obj.hashCode() == this.hashCode();
20      }
21      @Override
22      public String toString() {
23          return "Название␣объекта␣-␣" + this.getName();
24      }
25  }
```

Листинг 8: Alive.java

```
1   import java.util.ArrayList;
2
3   public abstract class Alive extends Inanimate{
4       private final ArrayList<Emotions> emotions = new ArrayList<>();
5
6       Alive(String name){
7           super(name);
8       }
9
10      public ArrayList<Emotions> getEmotions(){
11          return emotions;
12      }
13
14      public void addEmotion(Emotions emotion){
15          emotions.add(emotion);
16      }
17  }
```

Листинг 9: Guy.java

```
1   public class Guy extends Alive{
2       Guy(String name){
3           super(name);
4           System.out.println(name + "␣успешно␣создан.");
5       }
6
7       @Override
8       public void addAbility(Abilities ability) {
9           super.addAbility(ability);
10          System.out.println(getName() + "␣умеет␣" + ability + ".");
11      }
12
13      @Override
14      public void addEmotion(Emotions emotion) {
15          super.addEmotion(emotion);
16          System.out.println(getName() + "␣чувствует␣" + emotion + ".");
```

```
17      }
18
19      @Override
20      public int hashCode() {
21          return super.hashCode()+this.getName().hashCode();
22      }
23      @Override
24      public boolean equals(Object obj) {
25          return obj.hashCode() == this.hashCode();
26      }
27      @Override
28      public String toString() {
29          return "Имя␣-␣" + this.getName();
30      }
31  }
```

Листинг 10: Environment.java

```
1   public enum Environment {
2       EVERYWHERE ("везде"),
3       STREET ("на␣улице"),
4       RESTAURANT ("в␣ресторане");
5
6       private String title = "";
7       private final String type;
8
9       Environment(String type){
10          this.type = type;
11      }
12
13      Environment(String type, String title) {
14          this(type);
15          this.title = title;
16      }
17
18      public String getType() {
19          return type;
20      }
21
22      public String getTitle() {
23          return title;
24      }
25
26      public int uniqueNumber() {
27          return (type + title).hashCode();
28      }
29
30      @Override
31      public String toString(){
32          return type + "␣" + title;
33      }
34
```

```
35      public Boolean equals(Environment somewhere) {
36          return somewhere.uniqueNumber() == this.uniqueNumber();
37      }
38  }
```

4 Результат выполнения

Незнайка успешно создан.
Незнайка умеет смотреть на коротышек в ресторане.
Незнайка чувствует зависть.
Козлик успешно создан.
Козлик умеет смотреть на коротышек в ресторане.
Козлик чувствует зависть.
Коротышки успешно создан.
Коротышки умеет угощаться разными вкусными блюдами в ресторане.
Незнайка умеет ходить на улице.
Незнайка умеет бродить на улице.
Объект Зеркало успешно создан.
Зеркало умеет криво отражать физиономии.
Козлик чувствует радость.
Незнайка чувствует счастье.

5 Вывод

При написании лабораторной работы я изучила принципы SOLID, научилась работать с интерфейсами, нумераторами и абстрактными классами.