

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



УНИВЕРСИТЕТ ИТМО

Вариант №335129.3  
Лабораторная работа №4  
по дисциплине  
"Программирование"

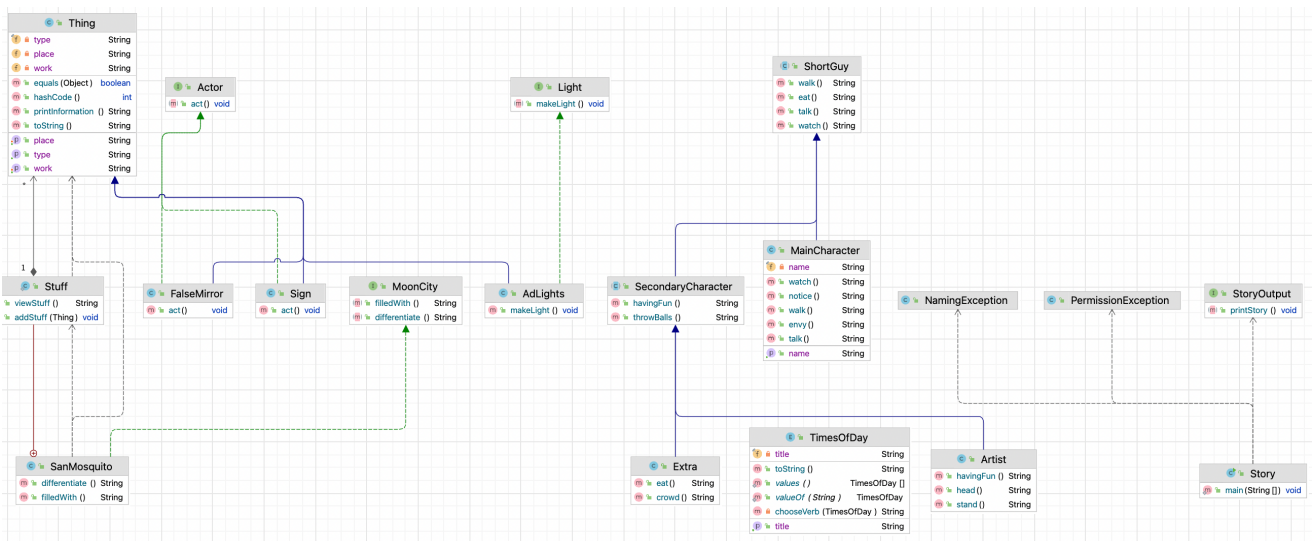
Выполнила студентка группы Р3117  
Галина Игнатова  
Преподаватель:  
Письмак Алексей Евгеньевич

г. Санкт-Петербург  
2021г.

# 1 Текст задания

Скоро наступил вечер. Повсюду засветились яркие огни реклам. Центральные улицы города, куда забрели в это время Незнайка с Козликом, наполнились электрическим светом, весельем и музыкой, гуляющими и танцующими коротышками, скрипением качелей, вертящихся каруселей, чертовых колес и других приспособлений для веселого времяпрепровождения. В этом отношении Сан-Комарик ничем не отличался от других больших лунных городов. 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. Посмеявшись, Незнайка заметил, что есть действительно стало хотеться меньше. В это время они увидели коротышек, собравшихся толпой перед небольшим деревянным помостом, над которым красовалась вывеска с надписью: "Веселый балаганчик". На помосте за занавеской, сделанной из обыкновенной простыни, стоял какой-то смешной коротышка. Он просунул голову в круглое отверстие, имевшееся посреди простыни, а стоявшие перед помостом зрители швыряли в него резиновыми мячами, целясь прямо в лицо. Коротышка смешно гримасничал и нелепо дергался в стороны, стараясь уберечь лицо от ударов, что очень веселило зрителей.

## 2 Диаграмма классов реализованной объектной модели



## 3 Исходный код программы

Листинг 1: Story.java

```
1 import City.SanMosquito;
2 import Time.TimesOfDay;
3 import ShortGuys.*;
4 import Things.*;
5 import Throwable.*;
6
```

```

7 public class Story {
8     public static void main(String[] args) {
9         try {
10             StoryOutput theStory = new StoryOutput() {
11                 @Override
12                 public void printStory() {
13                     System.out.println(TimesOfDay.Evening);
14                     System.out.println(new AdLights().printInformation());
15                     SanMosquito city = new SanMosquito();
16                     SanMosquito.Stuff.addStuff(new Thing("Электрический_свет"));
17                     SanMosquito.Stuff.addStuff(new Thing("веселье_и_музыка"));
18                     SanMosquito.Stuff.addStuff(new Thing("гуляющие_и_танцующие_
коротышки"));
19                     SanMosquito.Stuff.addStuff(new Thing("скрипение_качелей"));
20                     SanMosquito.Stuff.addStuff(new Thing("вертящиеся_карусели"));
21                     SanMosquito.Stuff.addStuff(new Thing("чертовы_колеса_и_другие_
приспособления_для_веселого_временипрепровождения"));
22                     System.out.println(city.filledWith());
23                     System.out.println(city.differentiate());
24                     System.out.println(new MainCharacter().watch());
25                     System.out.println(new Extra().eat());
26                     System.out.println(new MainCharacter().envy());
27                     System.out.println(new MainCharacter().walk());
28                     System.out.println(new FalseMirror().printInformation());
29                     System.out.println(new MainCharacter().talk());
30                     System.out.println(new MainCharacter().notice());
31
32                     System.out.println(new MainCharacter().watch());
33                     System.out.println(new Extra().crowd());
34                     System.out.println(new Sign().printInformation());
35                     System.out.println(new Artist().stand());
36                     System.out.println(new Artist().head());
37                     System.out.println(new Extra().throwBalls());
38                     System.out.println(new Artist().havingFun());
39                     System.out.println(new Extra().havingFun());
40                 }
41             };
42             StoryOutput theEnding = new StoryOutput() {
43                 @Override
44                 public void printStory() {
45                     System.out.println("Конец!");
46                 }
47             };
48             theStory.printStory();
49             theEnding.printStory();
50         }
51         catch(NamingException ex){
52             System.out.println("NamingException_caught:_ " + ex.getMessage());
53         }
54         catch(PermissionException ex){
55             System.out.println("PermissionException_caught:_ " + ex.getMessage());

```

```

56     }
57 }
58 }

```

Листинг 2: StoryOutput.java

```

1 public interface StoryOutput {
2     public void printStory();
3 }

```

Листинг 3: Thing.java

```

1 package Things;
2
3 public class Thing {
4     private final String type;
5     private String work;
6     private String place;
7
8     Thing(){
9         this.type = "Nothingness";
10    }
11
12    public Thing(String type){
13        this.type = type;
14    }
15
16    public String getType(){
17        return type;
18    }
19
20    public void setPlace(String place){
21        this.place = place;
22    }
23
24    public void setWork(String work){
25        this.work = work;
26    }
27
28    public String getPlace(){
29        return place;
30    }
31
32    public String getWork(){
33        return work;
34    }
35
36    public String printInformation(){
37        return getType() + "_" + getWork() + "_" + getPlace() + ".";
38    }
39
40
41    @Override

```

```

42     public int hashCode() {
43         return this.getType().hashCode();
44     }
45
46     @Override
47     public boolean equals(Object obj) {
48         return obj.hashCode() == this.hashCode();
49     }
50
51     @Override
52     public String toString(){
53         return getType();
54     }
55 }

```

Листинг 4: Actor.java

```

1  package Things;
2
3  public interface Actor {
4      public void act();
5  }

```

Листинг 5: Light.java

```

1  package Things;
2
3  public interface Light {
4      public void makeLight();
5  }

```

Листинг 6: AdLights.java

```

1  package Things;
2
3  public class AdLights extends Thing implements Light{
4      public AdLights(){
5          super("Огни_реклам");
6          makeLight();
7      }
8
9      @Override
10     public void makeLight() {
11         super.setWork("ярко_светят");
12         super.setPlace("всюду");
13     }
14 }

```

Листинг 7: Sign.java

```

1  package Things;
2
3  public class Sign extends Thing implements Actor{
4      public Sign(){
5          super("Вывеска_с_надписью_Весёлый\"_балаганчик\"");

```

```

6      act();
7  }
8
9  @Override
10 public void act(){
11     super.setWork("висела");
12     super.setPlace("над_деревянным_помостом");
13 }
14 }

```

Листинг 8: FalseMirror.java

```

1 package Things;
2
3 public class FalseMirror extends Thing implements Actor{
4     public FalseMirror(){
5         super("Кривое_зеркало");
6         act();
7     }
8
9     @Override
10    public void act(){
11        super.setWork("исказило_физиономии_смотрящих");
12        super.setPlace("у_краев_тротуара");
13    }
14 }

```

Листинг 9: MoonCity.java

```

1 package City;
2
3 import Things.Thing;
4
5 public interface MoonCity {
6     String filledWith();
7     String differentiate();
8 }

```

Листинг 10: SanMosquito.java

```

1 package City;
2
3 import Things.Thing;
4 import java.util.ArrayList;
5
6 public class SanMosquito implements MoonCity{
7     private final String name = "СанКомарик-";
8
9     public static class Stuff{
10         static ArrayList<Thing> CityStuff = new ArrayList<Thing>();
11         public static void addStuff(Thing thing){
12             CityStuff.add(thing);
13         }
14         public static String viewStuff(){

```

```

15         return CityStuff.toString().replace("[", "").replace("]", "");
16     }
17 }
18
19 @Override
20 public String filledWith(){
21     return Stuff.viewStuff() + "_заполнили_" + name + ".";
22 }
23
24 @Override
25 public String differentiate(){
26     return name + "_ничем_не_отличается_от_других_больших_лунных_городов.";
27 }
28 }

```

Листинг 11: NamingException.java

```

1 package Throwable;
2
3 public class NamingException extends RuntimeException{
4     NamingException(String line){
5         super(line);
6     }
7 }

```

Листинг 12: PermissionException.java

```

1 package Throwable;
2
3 public class PermissionException extends RuntimeException{
4     PermissionException(String message){
5         super((message.length() != 0) ? message : "Undefined");
6     }
7 }

```

Листинг 13: TimesOfDay.java

```

1 package Time;
2
3 public enum TimesOfDay {
4     Midnight("полночь"),
5     Noon("полдень"),
6     Morning("утро"),
7     Evening("вечер"),
8     Night("ночь"),
9     Dawn("рассвет"),
10    Twilight("сумерки");
11
12    private final String title;
13    TimesOfDay(String title){this.title = title;}
14
15    public String getTitle() {
16        return title;
17    }

```

```

18
19 private String chooseVerb(TimesOfDay time){
20     switch (time) {
21         case Midnight:
22         case Night:
23             return "Наступила_" + time.title + ".";
24         case Twilight:
25             return "Наступили_" + time.title + ".";
26         case Noon:
27         case Evening:
28         case Dawn:
29             return "Наступил_" + time.title + ".";
30         case Morning:
31             return "Наступило_" + time.title + ".";
32         default:
33             return "Наступило_какое-то_время_суток.";
34     }
35 }
36
37 @Override
38 public String toString(){
39     return chooseVerb(this);
40 }
41 }

```

Листинг 14: ShortGuy.java

```

1 package ShortGuys;
2
3 public abstract class ShortGuy {
4     ShortGuy(){}
5     public String walk(){
6         return "Коротышка_идёт.";
7     }
8     public String eat(){
9         return "Коротышка_ест.";
10    }
11    public String watch(){
12        return "Коротышка_смотрит.";
13    }
14    public String talk(){
15        return "Коротышка_говорит.";
16    }
17 }

```

Листинг 15: MainCharacter.java

```

1 package ShortGuys;
2
3 public class MainCharacter extends ShortGuy{
4     private final String name = "Незнайка_и_Козлик";
5     public MainCharacter(){
6         super();

```



```

7     }
8
9     public String getName(){
10         return name;
11     }
12
13     @Override
14     public String walk(){
15         return name + "_бродили_вдоль_улиц.";
16     }
17
18     @Override
19     public String talk(){
20         return name + "_засмеялись.";
21     }
22     @Override
23     public String watch(){
24         return name + "_смотрели_на_коротышек.";
25     }
26
27     public String notice(){
28         return name + "_заметили,_что_от_смеха_стало_легче.";
29     }
30
31     public String envy(){
32         return name + "_завидовали.";
33     }
34 }

```

Листинг 16: SecondaryCharacter.java

```

1 package ShortGuys;
2
3 public abstract class SecondaryCharacter extends ShortGuy{
4     SecondaryCharacter(){
5         super();
6     }
7
8     public String havingFun(){
9         return "Коротышки_веселились.";
10    }
11
12    public String throwBalls(){
13        return "Коротышки_кидались_мячами,_целясь_прямо_в_лицо.";
14    }
15 }

```

Листинг 17: Artist.java

```

1 package ShortGuys;
2
3 public class Artist extends SecondaryCharacter{
4     private final String name = "Смешной_коротышка";

```

```

5
6 public Artist(){
7     super();
8 }
9
10 @Override
11 public String havingFun(){
12     return name + "_гримасничал_и_нелепо_дергался_в_сторону,_стараясь_уберечь_лицо_
        от_ударов.";
13 }
14
15 public String stand(){
16     return name + "_стоял_на_помосте_за_занавеской,_сделанной_из_обыкновенной_
        простыни.";
17 }
18
19 public String head(){
20     return name + "_просунул_голову_в_круглое_отверстие,_имевшееся_посреди_простыни.
        ";
21 }
22 }

```

Листинг 18: Extra.java

```

1 package ShortGuys;
2
3 public class Extra extends SecondaryCharacter{
4     private final String name = "Коротышки";
5
6     public Extra(){
7         super();
8     }
9
10    @Override
11    public String eat(){
12        return name + "_наслаждались_вкусными_блюдами.";
13    }
14
15    public String crowd(){
16        return name + "_собрались_толпой.";
17    }
18 }

```

## 4 Результат выполнения

Наступил вечер.

Огни реклам ярко светят всюду.

Электрический свет, веселье и музыка, гуляющие и танцующие коротышки, скрипение качелей, вертящиеся карусели, чертовы колеса и другие приспособления для веселого времяпрепровождения заполнили Сан-Комарик.

Сан-Комарик ничем не отличается от других больших лунных городов.

Незнайка и Козлик смотрели на коротышек.

Коротышки наслаждались вкусными блюдами.  
Незнайка и Козлик завидовали.  
Незнайка и Козлик бродили вдоль улиц.  
Кривое зеркало исказило физиономии смотрящих у краев тротуара.  
Незнайка и Козлик засмеялись.  
Незнайка и Козлик заметили, что от смеха стало легче.  
Незнайка и Козлик смотрели на коротышек.  
Коротышки собрались толпой.  
Вывеска с надписью "Весёлый балаганчик" висела над деревянным помостом.  
Смешной коротышка стоял на помосте за занавеской, сделанной из обыкновенной простыни.  
Смешной коротышка просунул голову в круглое отверстие, имевшееся посреди простыни.  
Коротышки кидались мячами, целясь прямо в лицо.  
Смешной коротышка гримасничал и нелепо дергался в стороны, стараясь уберечь лицо от ударов.  
Коротышки веселились.  
Конец!

## 5 Вывод

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