

# Университет ИТМО

Программирование  
Отчет по лабораторной работе №3

ФИО студента: Готовко Алексей Владимирович

Номер варианта: 322040

Направление подготовки: 09.03.04 (СППО)

Учебная группа: Р3119

ФИО преподавателя: Письмак Алексей Евгеньевич

Санкт-Петербург

2021 г.

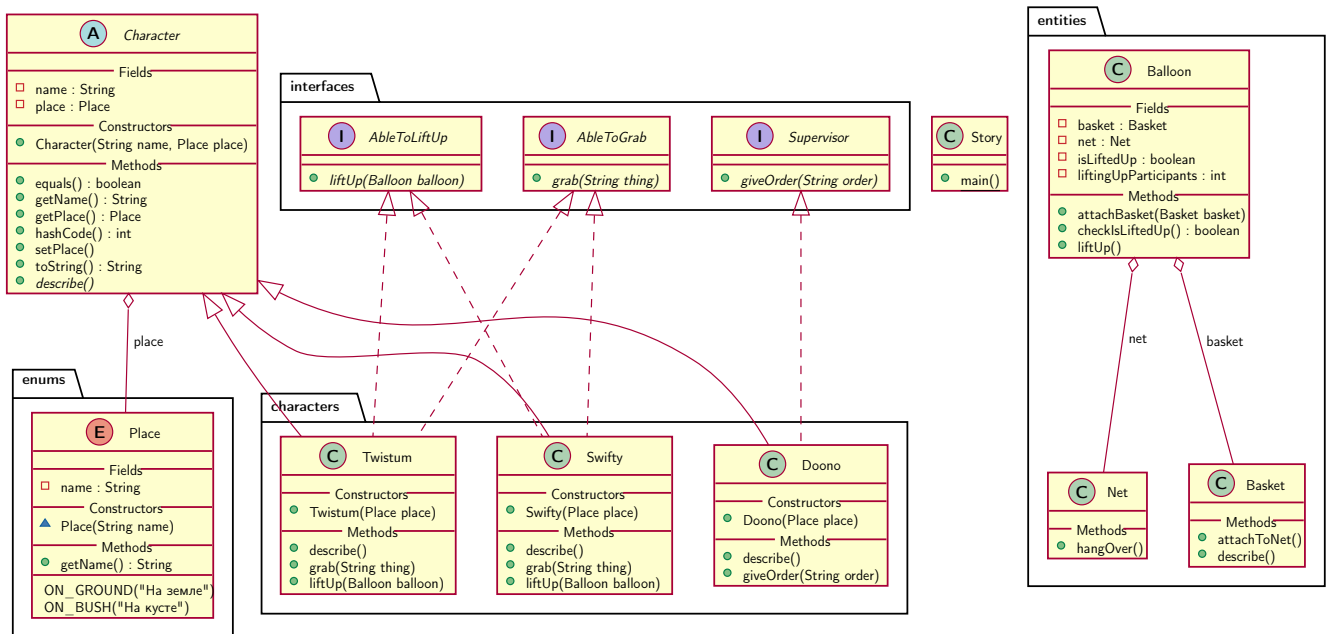
## Содержание

1	Текст задания	2
2	Диаграмма классов	2
3	Результат работы программы	3
4	Исходный код	3
5	Вывод	6

## 1 Текст задания

Сейчас же Торопыжка и Шпунтик взобрались с веревкой на куст и стали подтягивать шар кверху. Это очень обрадовало зрителей. Когда шар приподняли над землей, сетка по краям его свесилась вниз, и Знайка велел привязать к углам сетки корзину из березовой коры. Корзина была четырехугольная. С каждой стороны в ней было сделано по лавочке, и на каждой лавочке могло поместиться по четыре малыша.

## 2 Диаграмма классов



### 3 Результат работы программы

Персонажи:

Знайка – умный

Торопыжка – быстрый

Шпунтик – сообразительный

Торопыжка схватил предмет 'Веревка'

Шпунтик схватил предмет 'Веревка'

Торопыжка перешел в локацию 'На кусте'

Шпунтик перешел в локацию 'На кусте'

Торопыжка начинает приподнимать шар

Шпунтик начинает приподнимать шар

Шар приподнялся

Сетка свесилась вниз

Зрители обрадовались

Знайка велит привязать к углам сетки корзину из березовой коры

Теперь к сетке на шаре прикреплена корзина

Корзина из березовой коры. Четырехугольная. С каждой ее стороны есть лавочка, и на каждой лавочке может поместиться по четыре малыша

### 4 Исходный код

```
Character.java
1 import lab3.enums.Place;
2
3 abstract class Character {
4     private String name;
5     private Place place;
6
7     public Character(String name, Place place) {
8         this.name = name;
9         this.place = place;
10    }
11
12    public String getName() {
13        return this.name;
14    }
15
16    public Place getPlace() {
17        return this.place;
18    }
19
20    public void setPlace(Place place) {
21        this.place = place;
22        System.out.println(this.name + " перешел в локацию '" + place.getName() + "'");
23    }
24
25    abstract public void describe();
26
27    @Override
28    public String toString() {
29        return String.format("Character = {name: %s, currentPlace: %s}", this.name, this.place);
30    }
31
32    @Override
33    public boolean equals(Object obj) {
34        if (obj == null) return false;
35        if (obj instanceof Character) {
36            return (this.name.equals(((Character) obj).getName()) &&
37                this.place.equals(((Character) obj).getPlace()));
38        }
39        return false;
40    }
41
42    @Override
43    public int hashCode() {
44        return (this.name.hashCode() + this.place.hashCode());
45    }
46
47 }
48
```

Page 1 of 1

```
Doono.java
1 import lab3.enums.Place;
2 import lab3.interfaces.Supervisor;
3
4 public class Doono extends Character implements Supervisor {
5     public Doono(Place place) {
6         super("Знайка", place);
7     }
8
9     @Override
10    public void describe() {
11        System.out.println("Знайка -- умный");
12    }
13
14    @Override
15    public void giveOrder(String order) {
16        System.out.println("Знайка велит " + order);
17    }
18
19 }
```

Page 1 of 1

Swiftly.java

```

1
2 import lab3.entities.Balloon;
3 import lab3.enums.Place;
4 import lab3.interfaces.AbleToGrab;
5 import lab3.interfaces.AbleToLiftUp;
6
7 public class Swiftly extends Character implements AbleToGrab, AbleToLiftUp {
8     public Swiftly(Place place) {
9         super("Торопышка", place);
10    }
11
12    @Override
13    public void describe() {
14        System.out.println("Торопышка -- быстрый");
15    }
16
17    @Override
18    public void grab(String thing) {
19        System.out.println("Торопышка схватила предмет \"\" + thing + "\"");
20    }
21
22    @Override
23    public void liftUp(Balloon balloon) {
24        System.out.println("Торопышка начинает приподнимать шар");
25        balloon.liftUp();
26    }
27
28 }
29

```

Page 1 of 1

Twistum.java

```

1 import lab3.entities.Balloon;
2 import lab3.enums.Place;
3 import lab3.interfaces.AbleToGrab;
4 import lab3.interfaces.AbleToLiftUp;
5
6 public class Twistum extends Character implements AbleToGrab, AbleToLiftUp {
7     public Twistum(Place place) {
8         super("Шунтик", place);
9     }
10
11    @Override
12    public void describe() {
13        System.out.println("Шунтик -- сообразительный");
14    }
15
16    @Override
17    public void grab(String thing) {
18        System.out.println("Шунтик схватила предмет \"\" + thing + "\"");
19    }
20
21    @Override
22    public void liftUp(Balloon balloon) {
23        System.out.println("Шунтик начинает приподнимать шар");
24        balloon.liftUp();
25    }
26
27 }
28

```

Page 1 of 1

AbleToGrab.java

```

1 package lab3.interfaces;
2
3 public interface AbleToGrab {
4     public void grab(String thing);
5 }
6

```

Page 1 of 1

AbleToLiftUp.java

```

1 package lab3.interfaces;
2
3 import lab3.entities.Balloon;
4
5 public interface AbleToLiftUp {
6     public void liftUp(Balloon balloon);
7 }

```

Page 1 of 1

```

Supervisor.java
1 package lab3.interfaces;
2
3 public interface Supervisor {
4     public void giveOrder(String order);
5 }
6

```

Page 1 of 1

```

Place.java
1 package lab3.enums;
2
3
4 public enum Place {
5     ON_GROUND("На земле"),
6     ON_BUSH("На кусте");
7
8     private final String name;
9
10    Place(String name) {
11        this.name = name;
12    }
13
14    public String getName() {
15        return this.name;
16    }
17 }
18

```

Page 1 of 1

```

Balloon.java
1 package lab3.entities;
2
3 public class Balloon {
4
5     private Basket basket;
6     private Net net = new Net();
7     private boolean isLiftedUp = false;
8     private int liftingUpParticipants = 0;
9
10    public boolean checkIsLiftedUp() {
11        return isLiftedUp;
12    }
13
14    public void liftUp() {
15        liftingUpParticipants++;
16        if (liftingUpParticipants >= 2) {
17            isLiftedUp = true;
18            System.out.println("Шар приподнялся");
19            net.hangOver();
20        }
21    }
22
23
24    public void attachBasket(Basket basket) {
25        this.basket = basket;
26        basket.attachToNet();
27    }
28
29 }
30

```

Page 1 of 1

```

Basket.java
1 package lab3.entities;
2
3 public class Basket {
4
5     public void attachToNet() {
6         System.out.println("Теперь к сетке на шаре прикреплена корзина");
7         this.describe();
8     }
9
10    public void describe() {
11        System.out.println("Корзина из березовой коры. Четырехугольная. С каждой ее стороны есть лавочка, и на каждой лавочке может поместиться по четыре малыша");
12    }
13 }
14

```

Page 1 of 1

```

Net.java
1 package lab3.entities;
2
3 public class Net {
4
5     public void hangOver() {
6         System.out.println("Сетка свесилась вниз");
7     }
8 }
9
10

```

Page 1 of 1

```

Story.java
1 import lab3.entities.Balloon;
2 import lab3.entities.Basket;
3
4 import static lab3.enums.Place.*;
5
6 public class Story {
7     public static void main(String[] args) {
8         Doono doono = new Doono(ON_GROUND);
9         Swifty swifty = new Swifty(ON_GROUND);
10        Twistum twistum = new Twistum(ON_GROUND);
11
12        System.out.println("Начинаем:");
13        Character[] characters = {doono, swifty, twistum};
14        for (Character character : characters) {
15            character.describe();
16        }
17
18        Balloon balloon = new Balloon();
19        Basket basket = new Basket();
20
21        System.out.println();
22        swifty.grab("Вепешка");
23        twistum.grab("Вепешка");
24        swifty.setPlace(ON_BUSH);
25        twistum.setPlace(ON_BUSH);
26        swifty.liftup(balloon);
27        twistum.liftup(balloon);
28
29        if (balloon.checkIsLiftedUp()) {
30            System.out.println("Зрители обрадовались");
31        }
32
33        System.out.println();
34        doono.giveOrder("привязать к углам сетки корзину из березовой коры");
35        balloon.attachBasket(basket);
36    }
37 }
38

```

Page 1 of 1

## 5 Вывод

В результате выполнения лабораторной работы были изучены принципы SOLID, абстрактные классы, наследование, интерфейсы, тип данных enum, а также реализации методов класса Object по умолчанию: toString(), hashCode() и equals().