

Домашнее задание по теме
«Понятия ООП: наследование, полиморфизм»

Формулировка задания:

Задача 1. Расширить программу прошлого домашнего задания. Добавить механизм наследования.

В программе должно быть два класса – один для обычных продуктов – Product, а другой для специальных – DiscountProduct.

- Product - представляет обычный продукт из прошлого домашнего задания. Характеристики Продукта: название и стоимость. Название продукта не может быть пустой строкой, оно должно быть. Стоимость продукта не может быть отрицательным числом.

- Скидочный продукт — специальный продукт, цена которого снижена на размер скидки. У скидки есть также срок действия. После завершения срока действия скидка меняется.

Ограничения в классах для продуктов:

- Название продукта не должно содержать только цифры;
- Если название продукта короче, чем 3 символа, то такое название недействительно;
- Если стоимость продукта или скидочного продукта 0 или отрицательная, то такая цена неправильная. Должна быть ошибка валидации.

Программа реализуется в отдельной ветке `git homeworks/homework07`. При сохранении состояния программы (коммиты) пишется сообщение с описанием хода работы по задаче.

В корне папки с программой должен быть файл `.gitignore`.

Программа локально коммитится и публикуется в репозиторий GitHub на проверку.

<i>Тестовые данные:</i>	<i>Ожидаемый результат :</i>
Хлеб = 40 Молоко = 60 Торт = 800, 15% Кофе растворимый = 432, 50% END	Обычные продукты: Хлеб, Молоко Акционные продукты: Торт, Кофе растворимый
888 = 78 END	Недопустимое имя продукта!
ен = 78 END	Недопустимое имя продукта!
Шоколадка = 0 END	Недопустимая стоимость продукта!

```
1 package homeworks.homework07;
2
3 import java.util.Objects;
4
5 public class Product { 11 usages 1 inheritor new *
6     private String title; 7 usages
7     private Double cost; 6 usages
8     private Boolean allIsGood = true; 2 usages
9
10    public Product(String title, Double cost) { 2 usages new *
11        this.title = title;
12        this.cost = cost;
13    }
14
15    > public String getTitle() { return title; }
16
17
18    > public Double getCost() { return cost; }
19
20
21
22
23    public Boolean getAllIsGood() { 2 usages new *
24        return allIsGood;
25    }
26
27    public void setAllIsGood(Boolean allIsGood) { 2 usages new *
28        this.allIsGood = allIsGood;
29    }
30
```

@

```
57 @Override 1 override new *  
58 public int hashCode() { return Objects.hash(title, cost); }  
61  
62 @Override 1 override new *  
63 public String toString() { return title; }  
66 }
```

```
1 package homeworks.homework07;
2
3 import java.util.Objects;
4
5 public class DiscountProduct extends Product { 6 usages new *
6
7     private int discount; 4 usages
8     private boolean isOn = true; 6 usages
9
10    public DiscountProduct(String title, Double cost) { 1 usage new *
11        super(title, cost);
12    }
13
14    public int getDiscount() { no usages new *
15        return discount;
16    }
17
18    public boolean isOn() { no usages new *
19        return isOn;
20    }
21
22    public void setDiscount(int discount) { 1 usage new *
23        if (this.isOn) {
24            setCost(getCost() - discount * getCost() / 100);
25        }
26    }
27
```

```
28 public void setOn(boolean on) { no usages new *
29     isOn = on;
30 }
31
32 @Override new *
33 public boolean equals(Object o) {
34     if (o == null || getClass() != o.getClass()) return false;
35     if (!super.equals(o)) return false;
36     DiscountProduct that = (DiscountProduct) o;
37     return discount == that.discount && isOn == that.isOn;
38 }
39
40 @Override new *
41 public int hashCode() {
42     return Objects.hash(super.hashCode(), discount, isOn);
43 }
44
45 @Override new *
46 public String toString() {
47     return getTitle();
48 }
49 }
```



```
1 package homeworks.homework07;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5 import java.util.regex.Pattern;
6 import java.util.regex.Matcher;
7
8 public class App { new *
9     public static void main(String[] args) { new *
10
11         int indexNormal = 0;
12         int indexDiscount = 0;
13
14         Scanner scanner = new Scanner(System.in);
15         Pattern findDiscount = Pattern.compile(",");
16
17         Product[] product = new Product[0];
18         DiscountProduct[] discountProduct = new DiscountProduct[0];
19
20         while (true) {
21             String str = scanner.nextLine();
22             Matcher matcher = findDiscount.matcher(str);
23
24             if (str.equals("END")) {
25                 break;
26             } else {
```

```

27     if (!matcher.find()) {
28         product = Arrays.copyOf(product, product.length + 1);
29         product[indexNormal] = new Product("", (double) 0);
30         product[indexNormal].setTitle(str.split("=")[0]);
31         product[indexNormal].setCost(Double.parseDouble(str.
32             split("-")[1]));
33
34         indexNormal++;
35     } else {
36         discountProduct = Arrays.copyOf(discountProduct,
37             discountProduct.length + 1);
38         discountProduct[indexDiscount] = new DiscountProduct("",
39             (double) 0);
40
41         discountProduct[indexDiscount].setTitle(str.split("-")[0]);
42         discountProduct[indexDiscount].setCost(Double.parseDouble
43             (str.split("-")[1].split(",")[0]));
44         discountProduct[indexDiscount].setDiscount(Integer.parseInt
45             (str.split("-")[1].split(",")[1].replace("%", "")));
46
47         indexDiscount++;
48     }
49 }
50 }
51

```

```
52 String resultString = "";|
53
54 for (Product p : product) {
55     if (p.getAllIsGood()) {
56         resultString = resultString + p.getTitle() + ", ";
57     }
58 }
59
60 if (!resultString.equals("")) {
61     System.out.print("Обычные продукты: " + resultString.substring(0,
62         resultString.length() - 2));
63 }
64
65 resultString = "";
66
67 for (DiscountProduct dp : discountProduct) {
68     if (dp.getAllIsGood()) {
69         resultString = resultString + dp.getTitle() + ", ";
70     }
71 }
72
73 if (!resultString.equals("")) {
74     System.out.print("\nАкционные продукты: " + resultString.substring(0,
75         resultString.length() - 2));
76 }
77 }
78 }
```

```
"C:\Program Files\Java\jdk-21.0.2\bin\java.exe" -javaagent:G:\JetBrains\IntelliJ\Idea2025.1\lib\idea_rt.jar=57121 -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath D:\JavaProject\JavaDevHomeworks\out\production\JavaDevHomeworks homeworks.homework07.App
```

Хлеб=40

Молоко=60

Торт=800, 15%

Кофе растворимый=432, 50%

END

Обычные продукты: Хлеб, Молоко

Акционные продукты: Торт, Кофе растворимый

Process finished with exit code 0

```
"C:\Program Files\Java\jdk-21.0.2\bin\java.exe" -javaagent:G:\JetBrains\IntelliJ\Idea2025.1\lib\idea_rt.jar=57136 -Dfile.encoding=UTF-8 -Dsun.stdout
.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath D:\JavaProject\JavaDevHomeworks\out\production\JavaDevHomeworks homeworks.homework07.App
888=78
Недопустимое имя продукта!
END
Process finished with exit code 0
```

```
"C:\Program Files\Java\jdk-21.0.2\bin\java.exe" -javaagent:G:\JetBrains\IntelliJ\Idea2025.1\lib\idea_rt.jar=57144 -Dfile.encoding=UTF-8 -Dsun.stdout
.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath D:\JavaProject\JavaDevHomeworks\out\production\JavaDevHomeworks homeworks.homework07.App
ен=78
Недопустимое имя продукта!
END

Process finished with exit code 0
```

```
"C:\Program Files\Java\jdk-21.0.2\bin\java.exe" -javaagent:G:\JetBrains\IntelliJ\Idea2025.1\lib\idea_rt.jar=57154 -Dfile.encoding=UTF-8 -Dsun.stdout
.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath D:\JavaProject\JavaDevHomeworks\out\production\JavaDevHomeworks homeworks.homework07.App
```

```
Шоколадка=0
```

```
Недопустимая стоимость продукта!
```

```
END
```

```
Process finished with exit code 0
```