

## **«Работа с классами и объектами. IO»**

### Формулировка задания:

**Добавить работу с файлом для домашнего задания 6 «Понятия ООП: инкапсуляция».**

Характеристики Покупателя: имя, сумма денег и пакет с продуктами (массив объектов типа Продукт). Имя не может быть пустой строкой. Деньги не могут быть отрицательным числом.

Если Покупатель может позволить себе Продукт, то Продукт добавляется в пакет. Если у Покупателя недостаточно денег, то добавление не происходит.

Характеристики Продукта: название и стоимость. Название продукта не может быть пустой строкой, оно должно быть. Стоимость продукта не может быть отрицательным числом.

Данные Покупателей и Продукты считываются из файла. Продукты в цикле выбираются покупателями по очереди и, пока не введено слово END, наполняется пакет. Результат работы программы выводится в файл.

```
1 package homeworks.homework08;
2
3 import java.util.Arrays;
4 import java.util.Objects;
5
6 public class Person { 8 usages new *
7     private String name; 10 usages
8     private Double money; 9 usages
9     private String[] purchaseResult = new String[0]; 9 usages
10    private Product[] products = new Product[0]; 11 usages
11
12    public Person(String name, Double money) { 1 usage new *
13        this.name = name;
14        this.money = money;
15    }
16
17    > public String getName() { return name; }
18
19
20
21    > public Double getMoney() { return money; }
22
23
24
25    > public Product[] getProducts() { return products; }
26
27
28
29    public String[] getPurchaseResult() { 2 usages new *
30        return purchaseResult;
31    }
32
33    @ public void setName(String name) { 1 usage new *
34        if (!name.isEmpty()) {
35            this.name = name;
```

```

36     } else {
37         System.out.println("Имя не может быть пустым");
38     }
39 }
40
41 public void setMoney(Double money) { 1 usage new *
42     if (money < 0) {
43         System.out.println("Деньги не могут быть отрицательными");
44     } else {
45         this.money = money;
46     }
47 }
48
49 @ public void addProducts(Product product) { 1 usage new *
50
51     purchaseResult = Arrays.copyOf(purchaseResult, purchaseResult.length + 1);
52
53     if (this.money >= product.getCost()) {
54         products = Arrays.copyOf(products, products.length + 1);
55         products[products.length - 1] = product;
56         this.money = this.money - product.getCost();
57         purchaseResult[purchaseResult.length - 1] = this.name + " купил(-а) "
58             + product.getTitle();
59     } else {
60         purchaseResult[purchaseResult.length - 1] = this.name +
61             " не может позволить себе " + product.getTitle();
62     }

```

```
63 }
64
65 public void setPurchaseResult(String[] purchaseResult) { no usages new *
66     this.purchaseResult = purchaseResult;
67 }
68
69 @Override new *
70 public boolean equals(Object o) {
71     if (o == null || getClass() != o.getClass()) return false;
72     Person person = (Person) o;
73     return Objects.equals(name, person.name) && Objects.equals(money,
74         person.money) && Objects.deepEquals(products, person.products);
75 }
76
77 @Override new *
78 public int hashCode() {
79     return Objects.hash(name, money, Arrays.hashCode(products));
80 }
81
82 @Override new *
83 public String toString() {
84     if (products.length == 0) {
85         return name + " - Ничего не куплено";
86     } else {
87         return name + " - " + Arrays.toString(products);
88     }
89 }
90 }
```

```

1 package homeworks.homework08;
2
3 import java.util.Objects;
4
5 public class Product { 10 usages new *
6     private String title; 7 usages
7     private Double cost; 6 usages
8
9     public Product(String title, Double cost) { 1 usage new *
10         this.title = title;
11         this.cost = cost;
12     }
13
14     public String getTitle() { return title; }
15
16
17
18     public Double getCost() { return cost; }
19
20
21
22     public void setTitle(String title) { 1 usage new *
23         if (!title.isEmpty()) {
24             this.title = title;
25         }
26     }
27
28     public void setCost(Double cost) { 1 usage new *
29         if (cost > 0) {
30             this.cost = cost;
31         }
32     }
33

```

```
34  @Override new *
35  public boolean equals(Object o) {
36      if (o == null || getClass() != o.getClass()) return false;
37      Product product = (Product) o;
38      return Double.compare(cost, product.cost) == 0 && Objects.equals(title,
39          product.title);
40  }
41
42  @Override new *
43  public int hashCode() { return Objects.hash(title, cost); }
44
45
46
47  @Override new *
48  public String toString() { return title; }
49
50
51 }
```



```
1 package homeworks.homework08;
2
3 import java.io.*;
4 import java.util.Arrays;
5
6 public class App { new *
7     public static void main(String[] args) { new *
8
9         String[] buyers;
10        String[] productCost;
11        String[] initialData = new String[0];
12
13        try (BufferedReader bufferedReader = new BufferedReader(new FileReader(
14            "\\JavaProject\\JavaDevHomeworks\\src\\homeworks\\homework08" +
15                "\\input.txt"))) {
16
17            buyers = bufferedReader.readLine().split(";");
18            productCost = bufferedReader.readLine().split(";");
19
20            int index = 0;
21            String str = "";
22
23            while (str != null) {
24                str = bufferedReader.readLine();
25                initialData = Arrays.copyOf(initialData, initialData.length + 1);
26                initialData[index] = str;
27                index++;
28            }
```

```
29
30 } catch (IOException e) {
31     throw new RuntimeException(e);
32 }
33
34 Person[] person = new Person[buyers.length];
35
36 for (int i = 0; i < buyers.length; i++) {
37     person[i] = new Person("", (double) 0);
38     person[i].setName(buyers[i].split("=")[0].trim());
39     person[i].setMoney(Double.parseDouble(buyers[i].split("=")[1].trim()));
40 }
41
42 Product[] product = new Product[productCost.length];
43
44 for (int i = 0; i < productCost.length; i++) {
45     product[i] = new Product("", (double) 0);
46     product[i].setTitle(productCost[i].split("=")[0].trim());
47     product[i].setCost(Double.parseDouble(productCost[i].split("=")[1].trim()));
48 }
49
50 int index = 0;
51
52 while (true) {
53
54     if (initialData[index].equals("END")) {
```



```

55         break;
56     } else {
57         for (Person a : person) {
58             if (initialData[index].split("-")[0].trim().equals(a.getName())) {
59                 for (Product b : product) {
60                     if (initialData[index].split("-")[1].trim().
61                         equals(b.getTitle())) {
62
63                         a.addProducts(b);
64                     }
65                 }
66             }
67         }
68     }
69     index++;
70 }


71
72 try (BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(
73     "\\JavaProject\\JavaDevHomeworks\\src\\homeworks\\homework08" +
74     "\\output.txt"))) {
75
76     for (Person p : person) {
77         for (int i = 0; i < p.getPurchaseResult().length; i++) {
78             bufferedWriter.write(p.getPurchaseResult()[i] + "\n");
79         }
80     }

```

```
81
82     bufferedWriter.write("\n");
83
84     for (Person p : person) {
85         bufferedWriter.write(p.toString().replace("[", "").replace(
86             "]", "") + "\n");
87     }
88
89     } catch (IOException e) {
90         throw new RuntimeException(e);
91     }
92 }
93 }
94
```

```
"C:\Program Files\Java\jdk-21.0.2\bin\java.exe" -javaagent:G:\JetBrains\IntelliJ\Idea2025.1\lib\idea_rt
.jar=62458 -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath
D:\JavaProject\JavaDevHomeworks\out\production\JavaDevHomeworks homeworks.homework08.App
```

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

input.txt

×

+

—

□

×

ФайлИзменитьПросмотр

Павел Андреевич = 10000; Анна Петровна = 2000; Борис = 10  
Хлеб = 40; Молоко = 60; Торт = 1000; Кофе растворимый = 879; Масло = 150;  
Павел Андреевич - Хлеб  
Павел Андреевич - Масло  
Анна Петровна - Кофе растворимый  
Анна Петровна - Молоко  
Анна Петровна - Молоко  
Анна Петровна - Молоко  
Анна Петровна - Торт  
Борис - Торт  
Павел Андреевич - Торт  
END

Строка 1, столбец 1341 символ100%Windows (CRLF)UTF-8

output.txt

×

+

Файл

Изменить

Просмотр

Павел Андреевич купил(-а) Хлеб  
Павел Андреевич купил(-а) Масло  
Павел Андреевич купил(-а) Торт  
Анна Петровна купил(-а) Кофе растворимый  
Анна Петровна купил(-а) Молоко  
Анна Петровна купил(-а) Молоко  
Анна Петровна купил(-а) Молоко  
Анна Петровна не может позволить себе Торт  
Борис не может позволить себе Торт  
  
Павел Андреевич - Хлеб, Масло, Торт  
Анна Петровна - Кофе растворимый, Молоко, Молоко, Молоко  
Борис - Ничего не куплено

Строка 1, столбец 1

426 символов

100%

Unix (LF)

UTF-8