

 Evgen959 / Advanced_Backend Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) master ▾[Advanced_Backend](#) / [Lesen 027](#) / [code](#) / [hw_na_les027_1](#) / [src](#) / [Main.java](#) 

Evgen959 newLes27

3 minutes ago



83 lines (72 loc) · 3.69 KB

Code

Blame

Raw



```
1  /*
2  Предположим, вы пишете программу для on-line магазина У каждого товара есть
3  наименование, цена, рейтинг, количество штук на складе и т.д.
4  Ваша программа должна предлагать пользователю выбрать, как бы он хотел сортировать
5  товары и в зависимости от выбора пользователя, выводить список товаров в нужном
6  порядке
7  Например: по цене по возрастанию, по цене по убыванию, по рейтингу, по количеству
8  на складе.
9  */
10
11
12  import java.util.*;
13
14  public class Main {
15      public static void main(String[] args) {
16
17          /*String str1 = "Hello";
18          String str2 = "java";
19          System.out.printf("%s | %s | %s\n",str1,str2, str1);*/
20
21          Product p1 = new Product("Монитор", 179.99, 4.8, 10);
22          List<Product> products = new ArrayList<>(List.of(
23              new Product("Ноутбук", 2000, 4.8, 3),
24              new Product("Наушники", 119.99, 4.4, 38),
25              new Product("Стиральная машина", 840.50, 4.0, 1),
26              new Product("Телевизор", 410.50, 4.8, 6),
27              new Product("Ноутбук", 2100, 4.9, 8),
28              p1
29          )); // Ctrl + alt + v
30
31          ComparatorProduct[] comparators = {
32              new ComparatorProductByPrice(),
33              new ComparatorProductByQuantity(),
34              new ComparatorProductByTitle()
```

```
35         };
36
37         ProductUtil.print(products);
38         System.out.println("-----");
39
40         ComparatorProduct menu = menu(comparators);
41         Collections.sort(products, menu);
42         ProductUtil.print(products);
43
44         System.out.println("----- бес сканера -----");
45         ProductUtil.print(products);
46         System.out.println("-----");
47         Collections.sort(products, new ComparatorProductByTitle());
48         ProductUtil.print(products);
49         System.out.println("-----");
50         Collections.sort(products, new ComparatorProductByPrice());
51         ProductUtil.print(products);
52         System.out.println("-----");
53         Collections.sort(products, new ComparatorProductByPriceMax());
54         ProductUtil.print(products);
55         System.out.println("-----");
56         Collections.sort(products, new ComparatorProductByPriceMax());
57         ProductUtil.print(products);
58
59         // сортирует в обратном порядке
60         System.out.println("----сортирует в обратном порядке----");
61         Collections.sort(products, new ComparatorProductByPrice().reversed());
62         ProductUtil.print(products);
63         // сравнение по нескольким параметрам
64         System.out.println("----сравнение по нескольким параметрам----");
65         Collections.sort(products, new ComparatorProductByPrice()
66             .thenComparing(new ComparatorProductByTitle()));
67         ProductUtil.print(products);
68     }
69
70     public static ComparatorProduct menu(ComparatorProduct[] arrayComparators){
71         System.out.println("Выберите вариант сортировки:");
72         Scanner scanner = new Scanner(System.in);
73         for (int i = 0; i < arrayComparators.length; i++) {
74             System.out.printf("%d. %s\n", i+1, arrayComparators[i].getDescription());
75         }
76         int choice = scanner.nextInt()-1;
77         if (choice>=0 && choice < arrayComparators.length){
78             return arrayComparators[choice];
79         }
80         System.exit(0);
81         return null;
82     }
83 }
```