

 **Evgen959** / **Advanced\_Backend** Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [master](#) [Advanced\\_Backend](#) / [Homework](#) / [Les026\\_HW](#) / [les26\\_dz1](#) / [src](#) / [Main.java](#) **Evgen959** newHWLes26

10 minutes ago



72 lines (63 loc) · 3.62 KB

**Code**

Blame

Raw



```
1  /*
2      Предположим, вы пишете программу для on-line магазина У каждого товара есть наименование
3      цена, рейтинг, количество штук на складе и т.д.
4      Ваша программа должна предлагать пользователю выбрать, как бы он хотел сортировать товар
5      и в зависимости от выбора пользователя, выводить список товаров в нужном порядке
6      Например: по цене по возрастанию, по цене по убыванию, по рейтингу, по количеству на
7  */
8
9
10 import java.util.*;
11
12 public class Main {
13     public static void main(String[] args) {
14
15         List<Product> productList = new ArrayList<>();
16         productList.add(new Product("Fishing fod 1", 3, 11.55, 13));
17         productList.add(new Product("Fishing fod 2", 1, 5.06, 30));
18         productList.add(new Product("Fishing fod 3", 5, 24.98, 8));
19         productList.add(new Product("Fishing fod 4", 5, 20.58, 2));
20         productList.add(new Product("Fishing fod 5", 2, 14.00, 10));
21         ProductUtil.printListProduct(productList);
22
23         String[] sortingProducts = {
24             "By Price Ascending",
25             "By Price Descending",
26             "By Rating",
27             "By Quantity"
28         };
29
30         int number = scannerSorting(sortingProducts);
31         sorting(3, productList);
32
33         /*System.out.println("----- Comparator By Price Ascending -----")
34         Collections.sort(productList, new ComparatorProductByPriceAscending());
```

```
35         ProductUtil.printListProduct(productList);
36         System.out.println("----- Comparator By Price Descending -----");
37         Collections.sort(productList, new ComparatorProductByPriceDescending());
38         ProductUtil.printListProduct(productList);
39         System.out.println("----- Comparator By Rating -----");
40         Collections.sort(productList, new ComparatorProductByRating());
41         ProductUtil.printListProduct(productList);
42         System.out.println("----- Comparator By Quantity -----");
43         Collections.sort(productList, new ComparatorProductByQuantity());
44         ProductUtil.printListProduct(productList);*/
45     }
46
47     ✓ public static int scannerSorting (String[] sortingProducts){
48         Scanner scanner = new Scanner(System.in);
49         System.out.println("Выберите вид сортировки:");
50         for (int i = 0; i < sortingProducts.length; i++) {
51             System.out.println(i + " : " + sortingProducts[i]);
52         }
53         return scanner.nextInt();
54     }
55
56     ✓ public static void sorting (int number, List<Product> productList){
57         if (number == 0){
58             System.out.println("----- Comparator By Price Ascending -----");
59             Collections.sort(productList, new ComparatorProductByPriceAscending());
60         }if (number == 1){
61             System.out.println("----- Comparator By Price Descending -----");
62             Collections.sort(productList, new ComparatorProductByPriceDescending());
63         }if (number == 2){
64             System.out.println("----- Comparator By Rating -----");
65             Collections.sort(productList, new ComparatorProductByRating());
66         }if (number == 3){
67             System.out.println("----- Comparator By Quantity -----");
68             Collections.sort(productList, new ComparatorProductByQuantity());
69         }
70         ProductUtil.printListProduct(productList);
71     }
72 }
```