

Evgen959 / Advanced_Backend Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)[Advanced_Backend](#) / [Lesen 032](#) / [code](#) / [hw_na_les031_2](#) / [src](#) / [Main.java](#)

...



Evgen959 newHWTestMainLes32

43acba2 · 7 minutes ago



61 lines (51 loc) · 2.71 KB

Code Blame

Raw



```
1  /*
2  4. Дан список Student. Для каждого спортсмена задано имя и список оценок.
3  Список оценок это List<Integer>. Причем каждая оценка это число от 1 до 10,
4  у каждого студента не более 10 оценок. Т.е. максимальный балл,
5  который может набрать студент 100.
6  Напишите программу которая формирует строку вида: Jack - 19;John -23;Kate-31 где число означает,
7  сколько студент не добрал до максимального балла.
8  Причем, студенты должны идти в порядке возрастания "недобора".
9  */
10
11
12  import java.util.ArrayList;
13  import java.util.Collections;
14  import java.util.Comparator;
15  import java.util.List;
16
17  public class Main {
18      public static void main(String[] args) {
19
20          List<Student> studentList = new ArrayList<>();
21
22          studentList.add(new Student("Jack", 6, 9, 10, 2, 4, 8, 7));
23          studentList.add(new Student("John", 7, 9, 5, 2, 8, 10, 7, 3));
24          studentList.add(new Student("Kate", 2, 9, 10, 7, 4, 9, 7));
25          studentList.add(new Student("Ivan", 6, 9, 10, 2, 8, 7));
26          studentList.add(new Student("Olga", 6, 9, 8, 2, 4, 3, 7, 10, 10));
27
28          //Collections.sort(studentList, new ComparatorStudentByShortfall());
29          //System.out.println(studentList);
30          //System.out.println("-----");
31          //System.out.println(studentToString(studentList));
32
33          /*Student student1 = new Student("Jack");
34          student1.addGrade(8);
35          student1.addGrade(6);
36          student1.addGrade(4, 7, 10, 3, 6);
37          /**List<Integer> grades = student1.getGrades();
38          grades.add(100);**/
39          System.out.println(student1);*/
40      }
41
42      public static String studentToString(List<Student> studentList){
43          List<Student> sortedList = createSortedCopy(studentList, new ComparatorStudentByShortfall());
44          return listToString(sortedList);
45      }
```

```
46
47 ✓ private static String listToString(List<Student> sortedList) {
48     StringBuilder sb = new StringBuilder();
49     for (Student s: sortedList){
50         sb.append(s.getName()).append("-").append(s.getShortfall()).append(";");
51     }
52     sb.setLength(sb.length()-1); // обрезаем строку на один символ (;) в конце
53     return sb.toString();
54 }
55
56 ✓ public static List<Student> createSortedCopy(List<Student> list, Comparator<Student> comparator){
57     ArrayList<Student> result = new ArrayList<>(list);
58     Collections.sort(result, comparator);
59     return result;
60 }
61 }
```