

kotitehtävä 5 ron Ritari

```
*main.java × Measurable.java Paivamaara.java Ympyra.java PaivamaaraTe...
1 package song;
2
3 import java.util.ArrayList;
4 import java.util.Collections;
5
6
7
8 public class main {
9
10
11 public static void main(String[] args) {
12     ArrayList<song> list = new ArrayList<song>();
13
14     list.add(new song("slide", "barnacle boi", 2.48, 5.0));
15     list.add(new song("desire", "barnacle boi", 3.23, 3.5));
16     list.add(new song("mirrors", "memba", 2.28, 4.5));
17     System.out.println(list);
18
19     //sort by name
20     System.out.println("\nsort by name:");
21     Collections.sort(list, new Sortbyname());
22     list.stream().map(song -> song.getName()).forEach(System.out::println);
23
24     //sort by rating
25     System.out.println("\nsort by rating:");
26     Collections.sort(list);
27     list.stream().map(song -> song.getRating()).forEach(System.out::println);
28
29     //sort by artist
30     System.out.println("\nsort by artist:");
31     list.sort((song s1, song s2)->s1.getArtist().compareTo(s2.getArtist()));
32     list.stream().map(song -> song.getArtist()).forEach(System.out::println);
33
34     //sort by time
35     System.out.println("\nsort by time:");
36     Collections.sort(list, new Sortbytime());
37     list.stream().map(song -> song.getTime()).forEach(System.out::println);
38 }
39
40 }
41
```

```
song.java ×
1 package song;
2
3
4 import java.util.Comparator;
5
6
7 public class song implements Comparable<song> {
8
9     String name;
10    String artist;
11    double time;
12    double rating;
13
14    public song(String name, String artist, double time, double rating) {
15        super();
16        this.name = name;
17        this.artist = artist;
18        this.time = time;
19        this.rating = rating;
20    }
21
22    public song() {
23        // TODO Auto-generated constructor stub
24    }
25
26    public String getName() {
27        return name;
28    }
29    public void setName(String name) {
30        this.name = name;
31    }
32    public String getArtist() {
33        return artist;
34    }
35    public void setArtist(String artist) {
36        this.artist = artist;
37    }
38    public double getTime() {
39        return time;
40    }
41    public void setTime(double time) {
42        this.time = time;
43    }
44    public double getRating() {
45        return rating;
46    }
47    public void setRating(double rating) {
48        this.rating = rating;
49    }
50
51
```

```
song.java x
40 }
41 public void setTime(double time) {
42     this.time = time;
43 }
44 public double getRating() {
45     return rating;
46 }
47 public void setRating(double rating) {
48     this.rating = rating;
49 }
50
51
52 @Override
53 public String toString() {
54     return "song [name=" + name + ", artist=" + artist + ", time=" + time + ", rating=" + rating + "]";
55 }
56
57 @Override
58 public int compareTo(song o) {
59     if(getRating() < o.getRating()) {
60         return 1;
61     }
62     else if(getRating() > o.getRating()) {
63         return -1;
64     }
65     return 0;
66 }
67
68 }
69
70 class Sortbyname implements Comparator<song> {
71
72     // Method
73     // Sorting in ascending order of name
74 public int compare(song a, song b)
75 {
76
77     return a.name.compareTo(b.name);
78 }
79
80
81 }
82
83 class Sortbytime implements Comparator<song> {
84
85 @Override
86 public int compare(song o1, song o2) {
87     if(o1.getTime() < o2.getTime()) {
88         return 1;
89     }
90     else if(o1.getTime() > o2.getTime()) {
```

```
song.java x
56
57 @Override
58 public int compareTo(song o) {
59     if(getRating() < o.getRating()) {
60         return 1;
61     }
62     else if(getRating() > o.getRating()) {
63         return -1;
64     }
65     return 0;
66 }
67
68 }
69
70 class Sortbyname implements Comparator<song> {
71
72     // Method
73     // Sorting in ascending order of name
74 public int compare(song a, song b)
75 {
76
77     return a.name.compareTo(b.name);
78 }
79
80 }
81 }
82
83 class Sortbytime implements Comparator<song> {
84
85 @Override
86 public int compare(song o1, song o2) {
87     if(o1.getTime() < o2.getTime()) {
88         return 1;
89     }
90     else if(o1.getTime() > o2.getTime()) {
91         return -1;
92     }
93     return 0;
94 }
95
96 }
97
98 }
99 }
100
```

tulokset:

```
16 list.add(new song("mirrors", "memba", 2.28, 4.5));
74 // ...

Problems Javadoc Declaration Console x
[terminated> main (1) [Java Application] C:\Users\ronni\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.4.v20221004-1257\jre\bin\javaw.exe (2 Dec 2022, 13:37:17 - 13:37:17) [pid: 8092]
[song [name=slide, artist=barnacle boi, time=2.48, rating=5.0], song [name=desire, artist=barnacle boi, time=3.23, rating=3.5], song [name=mirrors, artist=memba, time=2.28, rating=4.5]]

sort by name:
desire
mirrors
slide

sort by rating:
5.0
4.5
3.5

sort by artist:
barnacle boi
barnacle boi
memba

sort by time:
3.23
2.48
2.28
```