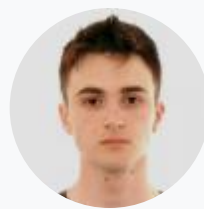




6000

Score: 15.000 (=100.0%)



Dominik  
Barukčić  
db53832@fer.hr  
0036538320

3 days

19:05:30

Done

Score: 2.500 (=100.0%)

Id: 41255

Pretpostavite da korisnik vašeg programa preko tipkovnice unosi podatke o studentima i ostvarenim bodovima na ispitu iz OOP-a, i to u sljedećem formatu:

```
Ime1#bodovi
Ime2#bodovi
```

Vaš zadatak je da dovršite klase Zapis i LabTask

```
class Zapis {
    String ime;
    Integer bodovi;

    public String toString() {
        return this.ime + ":" + this.bodovi;
    }

    // Dovršiti
}

class LabTask {
    public static Collection<Zapis> readData() {
        // Dovršiti
    }
}
```

tako da metoda **readData** s tipkovnice učita unesene podatke (dok se ne unese riječ "quit") i pohrani ih (prirodno) sortirano prema imenu u kolekciju koju vraća u pozivajuću metodu. Dva zapisa su **ISTI** ako imaju **isto ime**. Uočite da u kolekciji **ne smiju postojati duplikati**, tj. jedan student smije imati samo jedan broj bodova. Ako pri učitavanju podataka nađete na duplikat, u kolekciji mora ostati zapamćena **NOVA** vrijednost bodova.

Npr. neka korisnik s tipkovnice unese:

```
Sara#3
Ivan#5
Ana#7
Ana#8
Josip#5
quit
```

Kolekcija koju vraća metoda **readData()** treba sadržavati sljedeće podatke (sortirane prema imenu):

```
{Ana, 8}
{Ivan, 5}
{Josip, 5}
{Sara, 3}
```

gdje svaki redak predstavlja jedan objekt tipa **Zapis**.

Klasa **Zapis** mora sadržavati dvije članske varijable (*ime* i *bodovi*) te metodu *toString*, kako je navedeno u zadatku. Po potrebi može se proširiti s dodatnim varijablama i metodama (npr. može implementirati neko sučelje).

Prilikom predaje, potrebno je predati kompletne klase Zapis i LabTask. Iznad klasa možete dodavati importe iz standardnih Javinih biblioteka.

Student's answer:

```
1 import java.util.*;
2
3 class LabTask {
4     public static Collection<Zapis> readData() {
5         // Dovršiti
6         Collection<Zapis> col = new TreeSet<>();
7         Scanner sc = new Scanner(System.in);
8         while (sc.hasNextLine()) {
9             String line = sc.nextLine();
10            if(line.equals("quit")) {
11                break;
12            }
13            String[] data = line.split("#");
14            Zapis entry = new Zapis(data[0], Integer.parseInt(data[1]));
15            col.removeIf(element -> element.equals(entry));
16            col.add(entry);
17        }
18        return col;
19    }
20 }
21
22 class Zapis implements Comparable<Zapis>{
23     String ime;
24     Integer bodovi;
25
26     public String toString() {
27         return this.ime + ":" + this.bodovi;
28     }
29
30     // Dovršiti
31     public Zapis(String ime, Integer bodovi) {
32         this.ime = ime;
33         this.bodovi = bodovi;
34     }
35
36     @Override
37     public int compareTo(Zapis o) {
38         return this.ime.compareTo(o.ime);
39     }
40
41     @Override
42     public int hashCode() {
43         return super.hashCode();
44     }
45
46     @Override
47     public boolean equals(Object obj) {
48         if(!(obj instanceof Zapis other)) return false;
49         return this.ime.equals(other.ime);
50     }
51 }
```

Correct answer:

1 This test does not have "show solutions" option enabled.

Hint: Correct. Well done!

## Exam results obtained during submission evaluation:

| # | stdin   | stdout   | expected   | stderr | percentage | hint                      | mode  | correct? |
|---|---|--|--|--------|------------|---------------------------|---|----------|
| 1 | Gordan#3<br>Mario#4<br>Mihaela#5<br>Nikola#2<br>quit  | [Gordan:3, Mario:4, Mihaela:5, Nikola:2]   | [Gordan:3, Mario:4, Mihaela:5, Nikola:2]   |        | 100.00     | Correct.<br>Well<br>done! | check elements order :<br>false, case sensitive :<br>false, ignore whitespace :<br>true | true     |
| 2 | Gordan#3<br>Gordan#1<br>Gordan#1<br>Gordan#2<br>Mario#4<br>Mihaela#5<br>Nikola#2<br>Nikola#5<br>quit  | [Gordan:2, Mario:4, Mihaela:5, Nikola:5]   | [Gordan:2, Mario:4, Mihaela:5, Nikola:5]   |        | 60.00      | Correct.<br>Well<br>done! | check elements order :<br>false, case sensitive :<br>true, ignore whitespace :<br>true  | true     |
| 3 | Mihaela#5<br>Slavko#2<br>Nikola#2<br>Nikola#5<br>Vedran#4<br>Marko#5<br>Marko#4<br>Gordan#3<br>Mladen#1<br>Gordan#1<br>Gordan#1<br>Gordan#2<br>Mario#4<br>Gordan#1<br>Mislav#5<br>Nedjeljko#3<br>quit | [Gordan:1, Mario:4, Marko:4, Mihaela:5, Mislav:5, Mladen:1, Nedjeljko:3, Nikola:5, Slavko:2, Vedran:4] | [Gordan:1, Mario:4, Marko:4, Mihaela:5, Mislav:5, Mladen:1, Nedjeljko:3, Nikola:5, Slavko:2, Vedran:4] |        | 60.00      | Correct.<br>Well<br>done! | check elements order :<br>false, case sensitive :<br>true, ignore whitespace :<br>true  | true     |
| 4 | Sara#3<br>Ivan#5<br>Ana#7<br>Ana#8<br>Josip#5<br>quit   | [Ana:8, Ivan:5, Josip:5, Sara:3]   | [Ana:8, Ivan:5, Josip:5, Sara:3]   |        | 100.00     | Correct.<br>Well<br>done! | check elements order :<br>false, case sensitive :<br>true, ignore whitespace :<br>true  | true     |

Student's result

Correct result

Run student's code