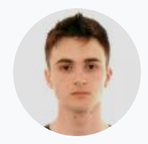
7000

Score: 17.500 (=100.0%)

1 2 3 4 5 6 7



Dominik
Barukčić
db53832@fer.hr
0036538320

4 days
23:51:55
Done

Score: 2.500 (=100.0%)

Id: 41260

Pretpostavite da postoji sljedeci kod:

```
/**
 * Sucelje specificira funkcionalnosti vezane uz statisticku obradu bodova studenata. Jedan student moze imati vise bodova
 *
 */
interface StudentScoreStatistics{
    /**
     *
     * Za studenta biljezi bodove. Mozete pretpostaviti da ce argumenti uvijek biti valjani (nece biti null).
     * Vazno: u praksi je moguce vise puta pozvati ovu metodu za istog studenta.
     * Npr.:
     *  stat.addStudentScores("Luka", 4, 1, 2);
     *  stat.addStudentScores("Luka", 1, 4);
     *
     *
     * @param student student za kojeg se biljeze bodovi
     * @param scores bodovi studenta
     */
    void addStudentScores(String student, Integer... scores);

    /**
     * @return Vraca kolekciju zabiljezenih studenata prema redoslijedu dodavanja. Vraca praznu kolekciju ako nema zabiljezenih studenata.
     */
    Collection<String> getInsertionOrderedStudents();

    /**
     *
     * @param student
     * @return vraca bodove za studenta, sortirane prema prirodnom poretku. Ako nema studenta metoda vraca null.
     */
    Collection<Integer> getNaturallySortedPointsForStudent(String student);

}
```

Vaš je zadatak napraviti klasu `Solution` koja će implementirati navedeno sučelje.

Opaske:

- klasu definirajte na razini package-private (dakle, bez modifikatora vidljivosti)
- po potrebi mozete dodati vlastiti import za kolekcije koje smatrate potrebnima

Student's answer:

```
import java.util.*;

class Solution implements StudentScoreStatistics {

    Map<String, List<Integer>> studentsScores;

    public Solution() {
        studentsScores = new LinkedHashMap<>();
    }

    @Override
    public void addStudentScores(String student, Integer... scores) {
        List<Integer> scoreList = new ArrayList<>(scores.length);
        Collections.addAll(scoreList, scores);
        studentsScores.put(student, scoreList);
    }

    @Override
    public Collection<String> getInsertionOrderedStudents() {
        return new LinkedHashSet<>(studentsScores.keySet());
    }

    @Override
    public Collection<Integer> getNaturallySortedPointsForStudent(String student)
        if(!(studentsScores.containsKey(student))) return null;
        List<Integer> scores = studentsScores.get(student);
        Collections.sort(scores);
        return scores;
    }
}
```

Correct answer:

```
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	functionality	OK	OK	EdgarLibrary v0.11.1	100.00	Correct. Well done!	check elements order : false, case sensitive : false, ignore whitespace : true	true

Student's result | Correct result

Rerun student's code