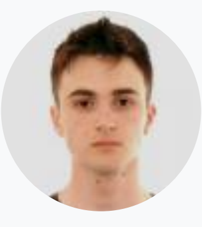
2 0 0 0

Score: 7.000 (=100.0%)

OOP-Lab5-Programski zadaci (178367)

1 ☺ 2 ☺



Dominik Barukčić
db53832@fer.hr
0036538320

01:45:22

Done

Score: 3.500 (=100.0%)
Id: 42299
Dostupna vam je klasa Superhero:

```
class Superhero {
    private String name;
    private String alias;
    private List<String> supervillains;

    public Superhero(String name, String alias, List<String> supervillains) {
        super();
        this.name = name;
        this.alias = alias;
        this.supervillains = supervillains;
    }

    public String getName() {
        return name;
    }

    public String getAlias() {
        return alias;
    }

    public List<String> getSupervillains() {
        return supervillains;
    }
}
```

Vaš je zadatak napisati klasu Solution koja sadrži dvije statičke metode: `Predicate<Superhero> superheroesFightingSupervillain(String supervillain)` i `Predicate<Superhero> superheroesWithOnlyOneWordInAlias()`.

Metoda `superheroesFightingSupervillain` vraća predikat koji omogućuje testiranje bori li se superheroj (`Superhero`) sa superzločincem čije je ime jednako kao primljeni `String`.

Metoda `superheroesWithOnlyOneWordInAlias` vraća predikat koji omogućuje testiranje ima li superheroj (`Superhero`) samo jednu riječ u svom aliasu.

Klasa `Solution` mora imati vidljivost postavljenu na package private.

Primjer korištenja:

```
Superhero captainAmerica = new Superhero("Steve Rogers", "Captain America",
        Arrays.asList(new String[]{"Red Skull", "Thanos"}));
Superhero hulk = new Superhero("Bruce Banner", "Hulk",
        Arrays.asList(new String[]{"Abomination", "Loki", "Thanos"}));
Superhero ironMan = new Superhero("Tony Stark", "Iron Man",
        Arrays.asList(new String[]{"Mandarin", "Loki", "Thanos"}));

List<Superhero> avengers = new ArrayList<Superhero>();
avengers.add(captainAmerica);
avengers.add(hulk);
avengers.add(ironMan);

System.out.println("Superheroes who are fighting a supervillain named Loki:");
Stream<Superhero> filtered1 = avengers.stream().filter(Solution.superheroesFightingSupervillain("Loki"));
for (Superhero superhero : filtered1.toArray(Superhero[]::new)) {
    System.out.println(superhero.getAlias());
}

System.out.println("\nSuperheroes who have only one word in their alias:");
Stream<Superhero> filtered2 = avengers.stream().filter(Solution.superheroesWithOnlyOneWordInAlias());
for (Superhero superhero : filtered2.toArray(Superhero[]::new)) {
    System.out.println(superhero.getAlias());
}
```

Očekivani ispis:

```
Superheroes who are fighting a supervillain named Loki:
Hulk
Iron Man

Superheroes who have only one word in their alias:
Hulk
```

Student's answer:

```
import java.util.function.Predicate;

class Solution {
    static Predicate<Superhero> superheroesFightingSupervillain(String supervillai
        return superhero -> superhero.getSupervillains().contains(supervillain);
    }

    static Predicate<Superhero> superheroesWithOnlyOneWordInAlias() {
        return superhero -> superhero.getAlias().split(" ").length == 1;
    }
}
```

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	functionalityFightingSupervillain	OK	OK	EdgarLibrary v0.11.1	50.00	Correct. Well done!	check elements order : false, case sensitive : false, ignore whitespace : true	true
2	functionalityOnlyOneWordInAlias	OK	OK	EdgarLibrary v0.11.1	50.00	Correct. Well done!	check elements order : false, case sensitive : false, ignore whitespace : true	true

Student's result | Correct result

Rerun student's code