REC-CIS



CS23333-Object Oriented Programming Using Java-2023

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-11-Set, Map / Lab-11-Logic Building

Quiz navigation



Show one page at a time Finish review Status Finished
Started Saturday, 16 November 2024, 7:35 PM
Completed Saturday, 16 November 2024, 8:03 PM
Duration 28 mins 38 secs

Question 1
Correct
Marked out of 1.00
Filag question

Java HashSet class implements the Set interface, backed by a hash table which is actually a HashMap instance.

No guarantee is made as to the iteration order of the hash sets which means that the class does not guarantee the constant order of elements over time.

This class permits the null element.

The class also offers constant time performance for the basic operations like add, remove, contains, and size assuming the hash function disperses the elements properly among the buckets.

Java HashSet Features

A few important features of HashSet are mentioned below:

- Implements Set Interface.
- The underlying data structure for HashSet is Hashtable.
- As it implements the Set Interface, duplicate values are not allowed.
- Objects that you insert in HashSet are not guaranteed to be inserted in the same order. Objects are inserted based on their hash code
- NULL elements are allowed in HashSet.
- HashSet also implements Serializable and Cloneable interfaces.

```
public class HashSet<E> extends AbstractSet<E> implements Set<E>, Cloneable, Serializable
Sample Input and Output:
5
90
56
45
78
25
78
Sample Output:
78 was found in the set.
3
2
7
9
5
Sample Input and output:
3
2
7
9
5
Sample Input and output:
5 was not found in the set.
```

Answer: (penalty regime: 0 %)

Reset answer

	,,	Expected	Got	
1	5	78 was found in the set.	78 was found in the set.	
	90			
	56			
	45			
	78			
	25			
	78			
2		${\bf 5}$ was not found in the set.	5 was not found in the set.	
	-1			
	2			
	4			
	5			

Question 2
Correct
Marked out of 1.00
F Flag question

Write a Java program to compare two sets and retain elements that are the same

Sample Input and Output:

5

Football Hockey

Cricket

Volleyball

Raskethall

```
Cricket
Badminton
Hockey
Volleyball
Handball
SAMPLE OUTPUT:
Football
Hockey
Cricket
Volleyball
Basketball
Answer: (penalty regime: 0 %)
1 | import java.util.HashSet;
2 | import java.util.Scanner;
          public class CompareSets {
               public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
                     int n1 = sc.nextInt();
                     sc.nextLine();
HashSet<String> set1 = new HashSet<>();
   10
11
12
                     for (int i = 0; i < n1; i++) {
    set1.add(sc.nextLine());</pre>
   13
                     }
   14
15
16
17
18
19
20
21
                     int n2 = sc.nextInt();
                     sc.nextLine();
HashSet<String> set2 = new HashSet<>();
                     for (int i = 0; i < n2; i++) {
    set2.add(sc.nextLine());</pre>
   22
23
24
25
26
27
                     set1.retainAll(set2);
                     for (String element : set1) {
    System.out.println(element);
}
   28
29
30
31
32
33
                     sc.close();
      Test Input
                           Expected Got
```

```
Cricket
                                       Cricket
                        Hockey Hockey
Volleyball Volleyball
           Football
           Hockey
Cricket
                        Football Football
           Volleyball
           Basketball
           Golf
           Cricket
           Badmintor
Football
           Hockey
Volleyball
           Throwball
           Toy
                         Car
                                       Car
           Car
           Auto
           3
Car
           Bus
Passed all tests!
```

```
Question 3
Correct
Marked out of 1.00

Flag question
```

```
Java HashMap Methods
```

containsKey() Indicate if an entry with the specified key exists in the map

 ${\color{blue} \textbf{containsValue()}} \ \textbf{Indicate if an entry with the specified value exists in the map}$

putlfAbsent() Write an entry into the map but only if an entry with the same key does not already exist

remove() Remove an entry from the map

replace() Write to an entry in the map only if it exists

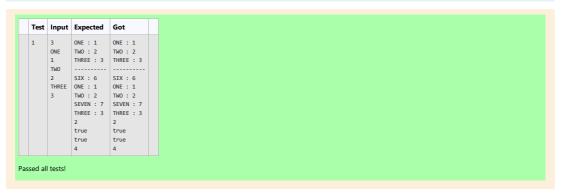
size() Return the number of entries in the map

Your task is to fill the incomplete code to get desired output

Answer: (penalty regime: 0 %)

7 // **HashSet 2:** Golf

```
20
21
                        map.put(name, num);
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
                   System.out.println("----");
                   // Creating another HashMap
                   HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
                   // Inserting key-value pairs to anotherMap using put() method anotherMap.put("SIX", 6); anotherMap.put("SEVEN", 7);
                   // Inserting key-value pairs of map to anotherMap using putAll() method anotherMap.putAll(map); // code here
                   // Printing key-value pairs of anotherMap
entrySet = anotherMap.entrySet();
for (Entry<String, Integer> entry : entrySet) {
    System.out.println(entry.getKey() + " : " + entry.getValue());
}
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
                   // Adds key-value pair 'FIVE-5' only if it is not present in map map.putlfAbsent("FIVE", 5);
                   // Retrieving a value associated with key 'TWO'
int value = map.get("TWO");
System.out.println(value);
                   // Checking whether key 'ONE' exists in map
System.out.println(map.containsKey("ONE"));
                   // Checking whether value '3' exists in map
                   System.out.println(map.containsValue(3));
                   // Retrieving the number of key-value pairs present in map
System.out.println(map.size());
                                                                                                                                                                                                                      ✓
```



Finish review

 ◄ Lab-11-MCQ
 Jump to...