1. Homework 7

Posted: October/3/2018
Due: October/14/2018 24.00

All homework solutions are due October/14/2018 24.00. I recommend to submit at least one version of all homework solutions long before due date.

1.1. Homework 7.1 (30 Points)

Objective: Implementing a HashSet using generics and exceptions

Grading:
Correctness: You can lose up to 40% if your solution is not correct
Quality: You can lose up to 80% if your solution is poorly designed
Testing: You can lose up to 50% if your solution is not well tested
Explanation: You can lose up to 100% if your solution if you can not explain your solution during the grading session

Homework Description:

You have to implement a subset of a set Your implementation must allow to insert the *null* element. Inserting an element must be done based onthe hash value of the element, with the exception of the null element.

Explanation:
Given is the following interface:

```
 1      public interface SetI<E> {
 2              boolean add(E e);
 3
 4              boolean addAll(SetI<? extends E> c);
 5              boolean containsAll(SetI<?> c);
 6              boolean removeAll(SetI<?> c);
 7              void    clear();
 8              boolean contains(Object o);
 9              boolean equals(Object o);
10              int     hashCode();
11              boolean isEmpty();
12              boolean remove(Object o);
13              int     size();
14              Object[]        toArray();
15              // <T> T[]      toArray(T[] a); try it, not required
16      }
```

Source Code: Src/27/SetI.java

You have to implement your class to match the following:

Your Work:

It might be useful

- to read the complete documentation before you start to design your solution.
- to understand what a hash value for a given object means.
- to understand what you do *if hashcode(o1) == hashcode(o2)* and *o1 != o2*.
- to think about in which order you should develop the methods
- to think about how you will test the methods.

You can not use any existing Java class for this home work.

Requirements:

You have to provide a test environment for your work. You have to name your class *MyHashSet.java*.

I will test your submission with the following test program:

Example:

An example of a solution execution:

THe following example is a small snippet of my code:

```
 public static void main(String args[] )      {
        SetI<String> aSet = new MyHashSet<String>();
        SetI<String> bSet = new MyHashSet<String>();

        String[] aStrings = { "a", "b", "c" };
        String[] bStrings = { "A", "B", "C" };
        aSet.add(aStrings[0]); aSet.add(aStrings[1]);           // setup a, b
        bSet.add(bStrings[0]); bSet.add(bStrings[1]);           // setup A, B

        System.out.println("aSet = " + aSet );                  // --> a, b

        for (int index = 0; index < aStrings.length; index ++ ) {       // contans a and b
                System.out.println("does " +
                                ( aSet.contains(aStrings[index]) ? "" : " not " ) + "conta
                                aStrings[index] );
        }
        System.out.println("aSet = " + aSet );                  // --> a, b

        System.out.println("aSet.remove(aStrings[0]); = " + aSet.remove(aStrings[0]) ); //
        System.out.println("aSet.remove(aStrings[2]); = " + aSet.remove(aStrings[2]) ); //
        System.out.println("aSet = " + aSet );

        aSet.addAll(bSet);                                      // --> b, A, B
        System.out.println("aSet = " + aSet );

        aSet.add(null);                                         // --> b, A, B, null
        System.out.println("aSet = " + aSet );
        System.out.println("aSet.remove(null); = " + aSet.remove(null) );       // can rem
  }
```

This code aboive produces the following output:

```
% java TestMyHashSet
aSet =  0:  0/null  1/null  2/null
    1:  0/null  1/null  2/null
    2:  0/null  1/null  2/null
    3:  0/null  1/null  2/null
    4:  0/null  1/null  2/null
    5:  0/null  1/null  2/null
    6:  0/null  1/null  2/null
    7:  0/a 1/null  2/null
    8:  0/b 1/null  2/null
    9:  0/null  1/null  2/null
```

```
does contain: a
does contain: b
does  not contain: c
aSet =  0:  0/null  1/null  2/null
    1:  0/null  1/null  2/null
    2:  0/null  1/null  2/null
    3:  0/null  1/null  2/null
    4:  0/null  1/null  2/null
    5:  0/null  1/null  2/null
    6:  0/null  1/null  2/null
    7:  0/a 1/null  2/null
    8:  0/b 1/null  2/null
    9:  0/null  1/null  2/null

aSet.remove(aStrings[0]); = true
aSet.remove(aStrings[2]); = false
aSet =  0:  0/null  1/null  2/null
    1:  0/null  1/null  2/null
    2:  0/null  1/null  2/null
    3:  0/null  1/null  2/null
    4:  0/null  1/null  2/null
    5:  0/null  1/null  2/null
    6:  0/null  1/null  2/null
    7:  0/null  1/null  2/null
    8:  0/b 1/null  2/null
    9:  0/null  1/null  2/null

aSet =  0:  0/null  1/null  2/null
    1:  0/null  1/null  2/null
    2:  0/null  1/null  2/null
    3:  0/null  1/null  2/null
    4:  0/null  1/null  2/null
    5:  0/A 1/null  2/null
    6:  0/B 1/null  2/null
    7:  0/null  1/null  2/null
    8:  0/b 1/null  2/null
    9:  0/null  1/null  2/null

aSet = null 0:  0/null  1/null  2/null
    1:  0/null  1/null  2/null
    2:  0/null  1/null  2/null
    3:  0/null  1/null  2/null
    4:  0/null  1/null  2/null
    5:  0/A 1/null  2/null
    6:  0/B 1/null  2/null
    7:  0/null  1/null  2/null
    8:  0/b 1/null  2/null
    9:  0/null  1/null  2/null
    9: null
aSet.remove(null); = true
```

Submission:

```
% ssh glados.cs.rit.edu # or use queeg.cs.rit.edu if glados is down
# password
# go to the directory where your solution is ...
% try hpb-grd lab7-1 ’All files required’
# you can see if your submission was successful:
# try -q hpb-grd lab7-1
```