Foundations of Computer Science
Exam dated 18 February 2025
Programming Test

PARAMETRIC SET
The set abstract data type, a container of elements that contains no duplicates, is defined with the following interface

```
public interface Set<E>
{
   /**
      Inserts the specified data item into this set if it is not null and not
      already present.
      @param item element to be added to this set.
      @return true if this set did not already contain the specified element.
   */
   boolean add(E item);

   /**
      Checks if this set contains the specified item.
      @param o the data item to be checked.
      @return true if this set contains the specified data item.
   */
   boolean contains(Object o);

   /**
      Returns true if this set is empty.
      @return true if this set is empty.
   */
   boolean isEmpty();

   /**
      Removes the specified data item from this set if it is contained.
      @param o the specified item.
      @return true if this set contained the specified element.
   */
   boolean remove(Object o);

   /**
      Returns the number of data item in this set.
      @return the number of data item in this set.
   */
   int size();

   /**
      Returns an array view of this set.
      @return an array containing the data items of this set.
   */
   Object[] toArray();
}
```

where E is the parametric data type for the elements in this set.
Code first the class **public class S<E> implements Set<E>** that implements a set where the constructor constructs an empty set.
Then code the following extended set
public class SE<E extends Comparable<E>> extends S<E> where
- public SE() constructs an empty extended set
- public boolean contains(Object o) overrides the superclass method using a recursive implementation.
- public boolean equals(Object o) compares for equality this extended set to the specified object.
- public Object[] toSortedArray() returns an array containing the data items of this extended set sorted in ascending order according to their natural order.

In the code evaluation, the following commands will be used

```
$ javac STester.java
$ java STester
$ javac SETester.java
$ java SETester
```

Where the STester and SETester test classes are below.

```
public class STester {
   public static void main(String[] args) {
      // define constant String array
      final String[] COUNTRIES = {"Russia","China","Vietnam","China","Japan"};
      final String[] OTHER_COUNTRIES = {"Korea","Russia","Japan"};
      // define test set
      Set<String> s1 = new S();

      // insert data items into test set
      for (String p : COUNTRIES) s1.add(p);               // test of add method

      // print set size
      System.out.println("SIZE: " + s1.size());          // test of size method

      // print set elements
      Object[] items = s1.toArray();                  // test of toArray method
      System.out.print("ITEMS: ");
      for (Object o : items) System.out.print(o + " ");
      System.out.println();

      // check contains
      System.out.print("CONTAINS: ");
      for (String s : OTHER_COUNTRIES) {
         if (s1.contains(s)) System.out.print(s + " OK - "); // test of contains
         else System.out.print(s + " KO - ");
      }
      System.out.println();

      // remove items
      System.out.print("REMOVE: ");                        // test of remove method
      int index = 0;
      while (!s1.isEmpty()) {
         System.out.print(items[index] +" ");
         s1.remove(items[index++]);
      }
      System.out.println();

      System.out.println("SIZE: " + s1.size());           // test of size method
   }
}
public class SETester {
   public static void main(String[] args) {
      // define constant String arrays
      final String[] COUNTRIES = {"Russia","China","Vietnam","China","Japan"};
      final String[] OTHER_COUNTRIES = {"Thailand","Laos","Vietnam","Cambodia"};

      // define test sets
      SE<String> s1 = new SE();
      SE<String> s2 = new SE();

      // insert data items into test sets
      for (String p : COUNTRIES) s1.add(p);
      for (String p : OTHER_COUNTRIES) s2.add(p);
      // print set size
      System.out.println("SIZE S1/2: " + s1.size() +"/" + s2.size());

      // print unsorted data items
      Object[] items = s1.toArray();
      System.out.print("S1 UNSORTED ITEMS: ");
      for (Object o : items) System.out.print(o + " ");
      System.out.println();

      items = s2.toArray();
      System.out.print("S2 UNSORTED ITEMS: ");
      for (Object o : items) System.out.print(o + " ");
      System.out.println();

      // print sorted data items
      Object[] sortedItems = s1.toSortedArray();// test of toSortedArray method
      System.out.print("S1 SORTED ITEMS: ");
      for (Object o : sortedItems) System.out.print(o + " ");
      System.out.println();

      sortedItems = s2.toSortedArray();
      System.out.print("S2 SORTED ITEMS: ");
      for (Object o : sortedItems) System.out.print(o + " ");
      System.out.println();

      // compare sets
      System.out.print("EQUALS: ");
      if (s1.equals(s2)) System.out.println("S1 EQUALS S2");  // test of equals
      else System.out.println("S1 DOES NOT EQUAL S2");

      // compare sets
      s2 = s1;
      if (s1.equals(s2)) System.out.println("S1 EQUALS S2");  // test of equals
      else System.out.println("S1 DOES NOT EQUAL S2");

      // contains
      System.out.print("CONTAINS: ");
      for (String s : OTHER_COUNTRIES)
      {
         if (s1.contains(s)) System.out.print(s + " OK - "); // test of contains
         else System.out.print(s + " KO - ");
      }
      System.out.println();
      System.out.println("S1 SIZE " + s1.size());
   }
}
```

The OKS.txt and OKSE.txt text files display printouts on standard output when the STester and SETester classes are run.
The following scores will be assigned in the code evaluation:
 S: add 6, contains 6, remove 6, toArray 5 (TOT 23),
 SE: contains if recursive 2, equals 2, toSortedArray 2 (TOT 6)
one point can be awarded for style and two points for asymptotic time complexity of the contains method of class S when better than O(n).
Code that reports compile-time errors is usually considered insufficient.
In the submitted code, the student may not
- use instance variables with access specifiers other than private
- add methods other than private
- use classes from the Java Platform API of packages other than java.lang.
Student files must contain a Java comment as the first line, including the student's first and last name, ID, date, and workstation number.

FOUNDATIONS OF COMPUTER SCIENCE

Name____ID._______________________

Date _______________________ Room:______ DEI work station no.___________________

Undersign one of the following statements:

□ I submit the following files for evaluation:
( )S.java ( ) SE.java ( ) other____

□ I withdraw from the programming test.

Signature____