Assignment: Interface Design

Due: Fri. Nov. 8 at 5pm.

An interface is a Java tool used to show the express a commonality of operations of between classes. This question concerns the two classes shown at the end of this document.

1. Identify the common operations of the two classes. Then, write an interface named Trio (in a file named Trio.java) that expresses the commonality of the two classes. You will want to use the fact that both the ArrayList and the LinkedList implement the interface java.util.List. To check your interface, make certain these two classes will still compile if you add "implements Trio" to the class declaration.
2. Write a new class that implements the Trio interface (named superTrioClass.java). This class should contain three Strings. ToList() should return a Vector<String> object. Max() should return the String that is at least as large as the other two, when compared using the compareTo() method of the String class.

What to turn in: Create the interface Trio.java file and the new class superTrioClass.java file. Create a TrioTest.java file. In this file, create the main function and test the implemented ToList and Max functions of the superTrioClass class. Take screenshot of your sample run.

Create a class diagram to display how these classes are related in the given program.

/***** Class IntegerTrio.java **********/**

```
import java.util.ArrayList;

/** A class to represent three integers. */
public class IntegerTrio
{
  private int a, b, c;

  /** Create and initialize a new object.
     @param x the first value
     @param y the second value
     @param z the third value
  */
  public IntegerTrio (int x, int y, int z)
```

```
  {
    a = x;
    b = y;
    c = z;
  }

  /** Create a list out of the trio's contents.
     @return the list
   */
  public ArrayList<Integer> toList ()
  {
    ArrayList<Integer> newList = new ArrayList<Integer>();
    newList.add(a);
    newList.add(b);
    newList.add(c);
    return newList;
  }


  /** Find and return the maximum of the three integers and
      return it as an object.
     @return the maximum as an object
   */
  public Integer max()
  {
    if (a >= b && a >= c) return a;
    if (b >= a && b >= c) return b;
    return c;
  }

}
```

/***** Class DoubleTrio.java **********/**

```
import java.util.LinkedList;

/** Objects of this class hold three double values. */
public class DoubleTrio
{
  private double d1, d2, d3;
```

```
  /** Create a new object, initialized with the given values. */
  public DoubleTrio (double alpha, double beta, double gamma)
  {
    d1 = alpha;
    d2 = beta;
    d3 = gamma;
  }

  /** Take the values in the object and put them in a list.
      @return the new list
   */
  public LinkedList<Double> toList ()
  {
    LinkedList<Double> dll = new LinkedList<Double>();
    dll.add(d1);
    dll.add(d2);
    dll.add(d3);
    return dll;
  }

  /** Find and return the maximum of the three doubles, as a
      wrapper.
      @return the maximum, wrapped
   */
  public Double max()
  {
    if (d1 >= d2) {
      if (d1 >= d3) {
        return d1;
      }
    } else if (d2 >= d3) {
      return d2;
    }
    return d3;
  }

}
```

Criterion	Details	Deductions
Classes	At minimum, we need the Trio.java, new class superTrioClass.java, and TrioTest.java files.	-10: interface declaration is incomplete -10: toList method implementation in the new class is incomplete -10: max method implementation in the new class is incomplete -10: TrioTest driver class was not created to test the interface/class
Code quality	Identifier names, class names, proper use of public/private, ample comments in main, etc.	-15: incoherent, inconsistent coding style -10: comments were not used throughout the code
Test	Note, whether the program compile and run	-30: The portion of the code of the assignment is a copy -10 x: The required java files not submitted -10 x: The program does not compile -5: screenshot of the program run was not attached -10: class diagram of the program was not attached with the submission