Programming Homework Assignment #6

Due Date: Mon., June 15 (11:59 PM)(upload the source .java files, and and output file, ALL ZIPPED)

Problem: Re-do Prog.HW#4. Use a CORRECT version of the RomanNumeral class (you MUST correct any errors mentioned in your HW#4 grade feedback)! Then CHANGE the following in the RomanNumeral class:

- have the class implement Comparable<RomanNumeral>
- override the compareTo method: **int compareTo(RomanNumeral romNum)**, so it returns < 0 if this' value < romNum's value, > 0 if this' value > romNum's value, BUT if the values are equal, return this' romNumStr's compareTo(passing romNum's romNumStr)
- in the constructor with the String parameter, throw an exception if the mutator with the String parameter throws an exception, otherwise do as specified on HW#4
- in the mutator method that has a String parameter, if the parameter is null OR if the method that converts the String to an int (method C. in HW#4) returns false, throw an exception , otherwise assign as specified on HW#4
- in the private method that converts the String to an int (method C. in HW#4), CHANGE THIS METHOD so it **RETURNS false** if the String is empty OR not a valid RomanNumeral, but returns true if valid and the String was converted to an int

Note: you may throw IllegalArgumentException or write your own Exception class.

Write a Java application program in which main is in a separate class and file from the RomanNumeral class (I'm calling it **Prog6**). In main, declare a RomanNumeral object variable, AND an ArrayList<RomanNumeral> variable, and then in main:

- Call a static method in this Prog6 class that returns an ArrayList<RomanNumeral> based on INPUT FROM A TEXT FILE (see 1. below) and assign the return value to a local ArrayList<RomanNumeral> variable.
- Call the static **sort** method in the **Collections** class, passing the ArrayList variable(why the RomanNumeral had to implement Comparable and override compareTo)
- Call a static method in this Prog6 class that writes all the elements in the ArrayList<RomanNumeral> TO AN OUTPUT FILE. (see 2. below)
- Get the middle element, (divide the size by 2 to get the middle index, don't worry if the size is even or odd), and display it as the "median Roman Numeral" to System.out.

1. In the method that returns an ArrayList<RomanNumeral>, instantiate a new ArrayList<RomanNumeral> (saving into a local ArrayList), read a filename for

the input file from the user after prompting the user. Try to open the file. If the file doesn't open, return an empty ArrayList. Otherwise, IN A WHILE LOOP, read one String at a time, instantiate a RomanNumeral passing the String, and add it to the ArrayList. PUT THE TRY BLOCK WITH CATCH CLAUSE with an ERROR MESSAGE IN THIS LOOP so it doesn't add the RomanNumeral if the constructor throws an exception. When all size RomanNumerals have been successfully added to the ArrayList, close the input file and return the ArrayList in a *return* statement.

2. In the method that writes all the elements in the ArrayList<RomanNumeral> TO AN OUTPUT FILE read a filename for the output file from the user after prompting the user. Try to open the file. If the file doesn't open, display an error message and return. If it opens, do the following:
    a. Get an Iterator<RomanNumeral> from the ArrayList and assign to a local variable.
    b. Traverse the ArrayList using the Iterator (as shown in the ArrayList and Iterator example in Lesson 10), and write the toString of each RomanNumeral returned by the Iterator's next method to the output file. (You don't need to write any labels this time.)

NOTE: YOU ARE **NOT ALLOWED** TO CONVERT THE ArrayList INTO AN ARRAY!!!! Also, as in HW#4, you're NOT allowed to convert any String into an array!

NO ARRAYS ALLOWED IN THIS PROGRAM!

Turn in using the test input files on Catalyst (and check with my test output files).

Don't forget the extra credit in CodeLab!