Programming Homework Assignment #3

Due Date: Sunday, Feb. 7, 11:55 PM (see Class Notes about this)
Upload the source files (.java files) with output (copied and pasted to the end of the main file) ZIPPED into one .zip file (see Catalyst under Week 5 for where to submit)

Problem:
Write a C++ program which changes the partial SortedList class into a SortedCircularLList class, but change THE PARTIAL CODE given in a HW3_JavaCodeFile in Catalyst (next to this assignment). ONLY change where highlighted in the HW3_JavaCodeFile! Also, include a Java class based on the following UML class diagram (and heading in the code file)

BoardMessage (see code file for heading)
- sequenceNum: int - messageLines: ArrayList<String>
+ BoardMessage() + BoardMessage(seq: int) + setSequenceNum(seq:int) + setOneLine(line: string) + getSequenceNum(): int + getNumLines(): int + compareTo(): int (overrides Comparable<T> compareTo) + toString(): String

More details about some of the BoardMessage methods:

- in each constructor, instantiate an ArrayList<> for the messageLines instance variable (or assign in the declaration)
- in the constructor with the int parameter, assign the parameter to the sequenceNum instance variable
- setSequenceNum assigns the parameter to the sequenceNum instance variable
- setOneLine adds the parameter String to the messageLines instance variable
- getSequenceNum returns the value of the int instance variable
- getNumLines returns the size of the instance variable ArrayList
- compareTo compare the sequence number ONLY (return an int < 0 if this' sequenceNum < parameter's sequenceNum, 0, if they're the same, and an int > 0 otherwise)
- toString() method will first put the sequence number in the return String (see test runs for examples), then concatenate each String in the ArrayList of Strings, putting a newline ('\n') at the end of each String (use a loop, try using the ArrayList's iterator)

Write main so it has a SortedCircularLList<BoardMessage> variable. Open an input file the same way you did in Prog. HW#1 (using the openInputFile method). If it opens, assign a new instance of a SortedCircularLList<> to the main variable, and read the file which has several sets of: sequence number, number of lines, several Strings (assume same number as number of lines). For each set of data, (in a loop) declare a BoardMessage passing the sequence number. As you read each String, call the BoardMessage mutator to "set" that line in the next available ArrayList element (use the loop counter for the line number). (Reading the whole file AND inserting into the SortedCircularList should be a separate static method in the main class called from main.)

To make sure you read and added to the SortedCircularLList correctly, call the SortedCircularLList's display() method. Then have main call the display messages method described below. Test the SortedCircularList's **getEntry()**, **remove()**, passing a BoardMessage with the same sequence # as the headPtr's, and passing a non-existent one, displaying the appropriate messages (should be separate method). Then call the display messages method again.

Required static method in the main class method (call from main after filling the list): loop for 40000 milliseconds total, but pause (use pause method given in the HW3 Code File in Catalyst) for 2000 milliseconds between BoardMessages. Use the SortedCircularList's iterator (call the list's startIterator() before the loop, then call the list's next to get and call that BoardMessage's toString method and display the String).

See test runs on Catalyst. Test your programs using the same test values to see if you get the same results.

DON'T USE RECURSION.

Extra Credit Problems (due the last day of the quarter!): Textbook (Data Abstraction & problem Solving by Carrano) 6th Ed. p. 371 #6, also THIS HW#3 but making the list a CIRCULAR DOUBLY LINKED LIST with dummy head and tail nodes.