Programming Homework Assignment #1

Due Date: Sunday, Jan. 17, 11:55 PM

- Upload the source files (.java files) with output (copied and pasted to the end of the main file) (see Catalyst under Week 2 for where to submit) Please submit ONE .zip or .rar file!

INCLUDE IN **EACH PROGRAM (in every programming assignment) comments with**: (remember, this is part of your grade)

-Name of program
-Programmer's name
-Current Date
-Computer operating system and compiler you are using
-Brief description of the program (1-5 sentences)
-Comments before each method in the style of the textbook
-Variable names and descriptions of what they represent (best to put these next to the variable declarations)

Problem:

Write a Java program for practicing using the Stack ADTs given on Catalyst. You are to write a class called PostfixExpression which includes the following:

Private instance variables:

- String for the whole postfix expression
- ArrayList of Strings for the tokenized postfix expression (only visible to this class)
- double (for the result of the expression)

Public methods:

- default constructor (assigns an empty String to the String instance variable, and a new ArrayList to the instance ArrayList variable)
- constructor with a String parameter: call the default constructor, then call the mutator for the String instance variable
- mutator for the String instance variable (whole expression), which not only assigns the parameter to the String instance variable, but also calls the private *tokenize* method, then the private *evaluate* method
- (NO mutator for the double instance variable nor ArrayList)
- accessor for the String instance variable (return a String)
- accessor for the double instance variable
- (NO ACCESSOR FOR THE ARRAYLIST)

- method to return a String with the equivalent of the postfix expression as an infix expression USING AN ArrayStack, but DON'T SAVE THE INFIX EXPRESSION STRING, just return it as a String value (OR return "Invalid expression" if the stack is empty when it should NOT be). Change the ArrayStack class so the array size is 15 and the pop() method is made more efficient. Use the Algorithm given in the HW1_JavaCodeFile.

- AND the following private methods:
  - method given in the HW1_JavaCodeFile to tokenize the input String, assigning to the instance variable that's a ArrayList of Strings (see details in A. below)
  - method (NO PARAMETERS) to evaluate the postix expression (stored in the ArrayList of Strings), USING A local LinkedStack storing the result in the double instance variable . The LinkedStack class is given on Catalyst (generic type is double). Use the Algorithm given in the HW1_JavaCodeFile.

In main, declare a PostfixExpression variable and assign a default instance. After trying to open the file, if it opens, read a line, then process each one (in a loop) the following way until the end of file:

- call the PostfixExpression mutator for setting the String, passing the input String
- call the PostfixExpression method to get the infix String
- call the accessors for the PostfixExpression to get the String and the double
- display the input (postfix) String, infix String expression and the result returned by the accessors

Use public static method given in the HW1_CodeFile (call from main) that opens a file FOR INPUT using a filename input from the user and returns a Scanner object. In main, display an error message if the file doesn't open and end the program.

Hint: see Class Notes for 01/12/2016 for examples of ArrayLists and converting from a String to a number

See test runs on Catalyst. Test your programs using the test input files. Use the HW1 Input File.txt given on Catalyst.

DON'T USE RECURSION.

Extra Credit Problems (due the last day of the quarter!): Textbook (Data Structures and Abstractions with Java by Carrano) 4th Ed. pp. 177-178 #3, #5