Project 4

Stack Application

CS 240 – 02

2/1/2016

Alexandra Klimenko

[Klimenko.alexandra99@gmail.com](mailto:Klimenko.alexandra99@gmail.com)

Section 1. Project Specification:

The purpose of the project is to convert an infix expression into a prefix and postfix expressions entered by the user. It will keep asking for user’s input until the user hit enter. The assumption is that the user will only enter single character variables and no spaces. However, if the user misses a parenthesis, or types in two operators close in wrong order it will display error message and ask for input again..

A constructor is made for printing and asking the user for input, it is called in the main method. There are two methods for converting one convertPostfix and convertPrefix, both of them use the Java Stack Implementation and convert the input string into a character array in order to loop through each character and implement the algorithm. Also, I implemented a few supplemental methods such as isOperand, isOperator, isPreviousOperator, isLowerPrecedence, and isHigherPrecedence in order to make the program easier to read.

Section 2. Testing Methodology:

Firstly, I implemented the postfix method without taking in consideration the parenthesis. When the output was right I included them too and implemented the prefix method. In order to check for the correct input, I used different boundary expressions.

1. Letters(upper, lower, and numbers)

I made the program work for lower case letters only, but it was easy to change by adding two more or conditions in the isOperand method.

1. Parenthesis Check

In order to make sure that the user entered wrong number of parenthesis, I created a counter in the checkInput method to increment whenever user enters an opening parenthesis and decrement when user enters a closing parenthesis.

1. Continue Parenthesis Check

To make sure that the user does not enter a closing parenthesis right after an opening one I checked in the conversion methods that whenever a user enter a closing parenthesis to check if the last element in stack is an opening parenthesis.

1. Wrong operator order

If two operators are next to each other, there will be more or same amount of operators as operands. This is why I created two variables to count the operands and operators to see if there is a wrong input.

Section 3. Lessons Learned:

For future improvements I could calculate the expression.