Computer Science Department - San Francisco State University

CSC 413 – Fall 2017

**Gabriel Equitz – id: 915254839**

Assignment 1 – Expression Evaluator and Calculator GUI

Repository at: https://github.com/CSC-413-SFSU-02/csc413\_02\_p1-gequitz

**Introduction**: The project consisted of writing a computer program that works as a calculator for integer numbers and obeys the proper priority of mathematical operations. The program was written in Java in the Netbeans environment and has four classes:

1. **EvaluatorUI.java** – This class implements the calculator GUI interface, which has 20 buttons, consisting of numbers 0 to 9, the operators +, -, \*, /, ^, ( , ) and buttons to clear one symbol (C) another one to clear the entire expression (CE).
2. **Evaluator.java** – This class creates and populates two stacks, one for operands and another for operators. The class pushes and pops the operators and operands in such a way that the mathematical expression is evaluated with a correct order of operator priority. Two tokens that are not part of mathematical expressions are used by the computer code as markers, which make the code more efficient. One is the pound sign (#) which is placed at the bottom of the operator stack and the other is the exclamation sign (!) which placed at the end of the mathematical expression.
3. **Operand.java** – This class parses the mathematical expression in its individual numbers and has a method to return their values.
4. **Operator.java** – This is an abstract class. Subclasses of this class code the priority of the operators and how to perform the operations (sum, multiplication, etc.). This class contains an instance of a HashMap which uses keys as tokens and the values are instance of the operators.

To compile the project go to the directory where the files are and type the following:

javac \*.java

to run:

java EvaluatorUI Evaluator Operand Operator

to create a jar file:

jar –cvfe EvaluatorUI.jar EvaluatorUI \*.class

To run the jar file:

java –jar EvaluatorUI.jar

The diagram below shows how the classes interact with each other:

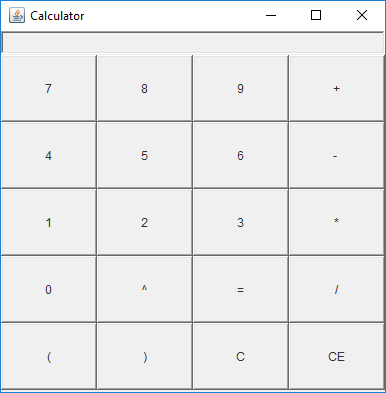
Operator

Operand

Evaluator

EvaluatorUI

The screenshot of the GUI calculator is shown below:



The biggest challenge I faced was to implement the parenthesis (round braces), but that was an optional feature in the project, and is not currently implemented.

Enhancements of this project could be the implementation of the parenthesis, implementing decimal digits and performing more advanced mathematical calculations, such as trigonometric functions, etc.

A more detailed structure of the classes and their respective methods is shown below:

File **EvaluatorUI**:

Class EvaluatorUI: implements the GUI interface

*Creates object of the type Evaluator*

*Methods*:

EvaluatorUI – constructor

Main(string argv[])

actionPerformed (ActionEvents arg0)

File **Evaluator**:

*Methods:*

Evaluator – constructor

Int eval(String expression) - calls Operator and Operand to calculate the mathematical functions.

File **Operand**:

*Methods:*

Operand (String token) - constructor

Operand (int Value) – constructor

boolean check (String token) – returns true if the specified token is an operand

int getValue() – returns the integer value of this operand

File **Operator**:

Operator: abstract Super Class

*Methods*:

boolean check (String token) – returns true if the specified token an operator

abstract int priority()

abstract Operand execute (Operand OperandOne, Operand OperandTwo ) –

Sub Classes of Operator:

PoundOperator, ExclamationOperator, AdditionOperator, SubtractionOperator, MultiplicationOperator, DivisionOperator, ExponentiationOperator.

Each of these classes having two methods:

Int priority : returns the priority of the operation

Operand execute : performs mathematical operations