CSC 413 Project Documentation

Fall 2018

Grant Kennedy

913056513

413.02

https://github.com/csc413-02-fa18/csc413-p1-grantwkenn.git

Table of Contents

1 Introduction

1.1 Project Overview

This project implements an integer calculator to calculate infix integer expressions with some basic mathematical operators.

1.2 Technical Overview

The calculator backend function Evaluator.java takes an infix expression from the user interface as a String of tokens. It parses each token in the expression and stores operators and operands in their respective stacks. When an expression is processed, the result is pushed as an operand to the top of its stack. The function manages when to process expressions based on the relative priority of each expression. After each expression has been processed, the evaluator stack is empty, and the result of the infix expression is left in the operand stack, to be returned to the front end user interface for display as an integer.

1.3 Summary of Work Completed

Implemented the evaluation algorithm in Evaluator.java using a helper method processExpressions(). Extended subclasses of Operator for each mathematical operator used. Connected evaluation to a provided user interface, with exception handling to display a blank screen result in case of user error or impossible expressions.

2 Development Environment

This project was developed with Java version: 1.8.0_161 in the Eclipse IDE for Java Developers with Build Version: Photon Release (4.8.0). Eclipse was installed on a PC with Windows 10.

3 How to Build/Import your Project

Download this program using the following GitHub link:

https://github.com/csc413-02-fa18/csc413-p1-grantwkenn.git

The project was imported into Eclipse from the file system by importing the master folder titled “csc413-p1-grantwkenn”. The project was built automatically throughout development in the Eclipse IDE. With “Build Automatically” disabled, the project can be built with “Ctrl+B” or from the “Project” dropdown menu.

4 How to Run your Project

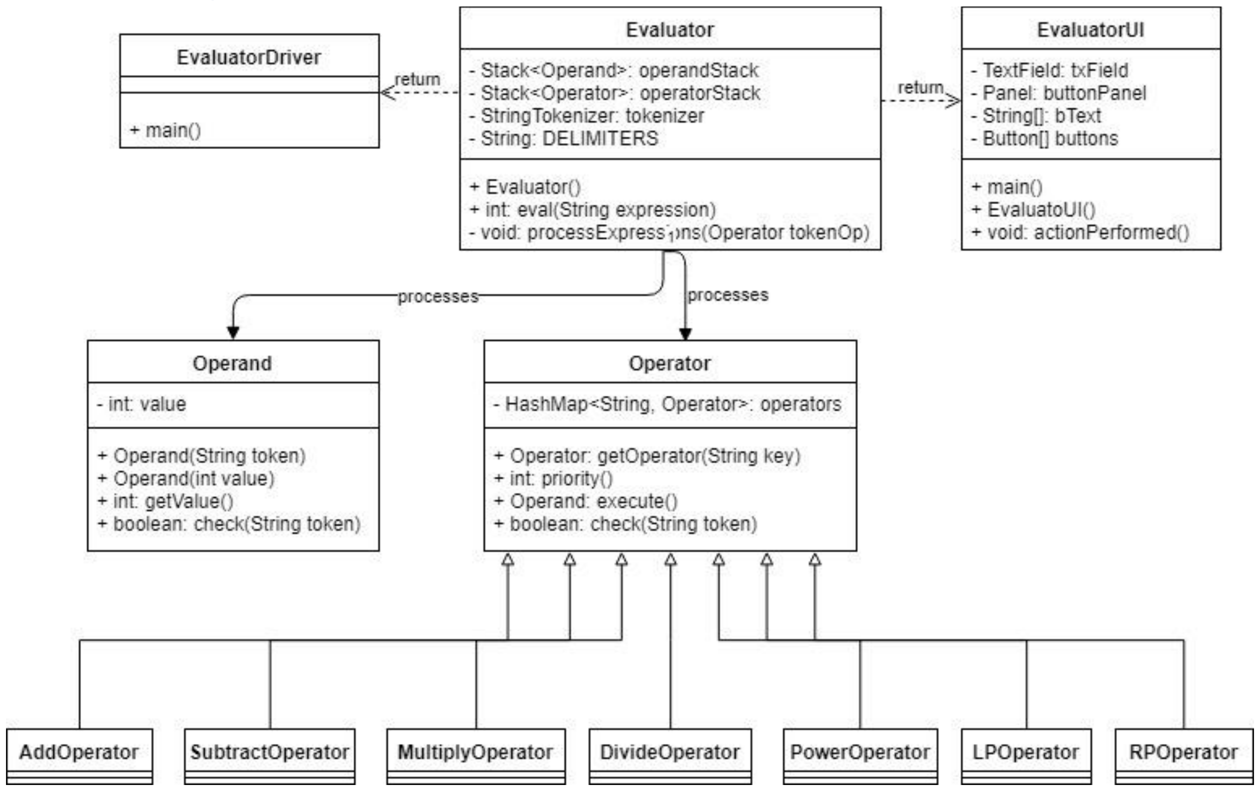
Project was run from within the Eclipse IDE by pressing Ctrl+F11 or from the “Run” dropdown menu.

5 Assumption Made

This calculator uses integer variables, therefore expressions such as (3/10) are evaluated to 0. Additionally, this calculator gives unpredictable results if expressions are too large. For example, the expression (10^10) gives an unpredictable result. However, using the Eclipse compiler, this program was able to give a reliable result for the expression (10^9). Most user errors will cause a blank screen output, but some may produce unexpected results. No user error should cause the program to crash.

6 Implementation Discussion

6.1 Class Diagram



7 Project Reflection

This project was implemented with the expected results. During development, it was clear from the algorithm how to satisfy each condition for each operator in the expression, but what took the most time was to consolidate the code into fewer lines by using the member function processExpressions(), as well as merging conditions together with if statements, and skipping appropriate parts of the loop for certain conditions to reduce unnecessary code. A few edge cases for very complex expressions required more time and debugging to explore how the algorithm was handling these edge cases, especially expressions using multiple parentheses. The error found in these edge cases were fixed by a few extra checks in the processExpressions() method to handle parentheses properly. The Evaluation driver main method helped with debugging to test many edge cases and use print statements for some basic debugging.

There is probably a more efficient implementation of the eval function which could have made this a better program. Additionally, the CE functionality of the calculator UI could have probably been more efficiently or neatly implemented than in this iteration, although its efficiency is sufficient enough for this program.

8 Project Conclusion/Results

This project successfully implemented an infix expression calculator underneath a barebones user interface. The calculator can handle any arrangement of expressions with any combination of parentheses and operations provided that the result of any expression is not too large (see 5. Assumption Made). This program passed all test cases provided in the testing folders.