

CSC 413 Project Documentation

Fall 2018

Corey Russ

917717871

413-03

**[https://github.com/csc413-03-
fall2019/csc413-p1-scrable](https://github.com/csc413-03-fall2019/csc413-p1-scrable)**

Table of Contents

1	Introduction	3
1.1	Project Overview	3
1.2	Technical Overview	3
1.3	Summary of Work Completed	3
2	Development Environment.....	3
3	How to Build/Import your Project	3
4	How to Run your Project.....	4
5	Assumption Made	4
6	Implementation Discussion.....	4
6.1	Class Diagram	4
7	Project Reflection.....	4
8	Project Conclusion/Results	5

1 Introduction

1.1 Project Overview

The purpose of this project is to create a functioning calculator. Each expression is evaluated by using basic mathematics priority of operations. The mathematical expression is executed from the beginning to the end using these priorities.

1.2 Technical Overview

The aim of this project is to implement a functioning calculator by utilizing a hashmap and two stacks. Operators are used as keys in the hashmap and instances of the operators are the values. The two stacks are of type operand and type operator. The LIFO feature of stacks allows us to process operators and operands in the order they appear, as well as process by precedence of operators.

1.3 Summary of Work Completed

In order to get the program skeleton functioning, I implemented several operator classes which each represent their individual operator. Each operator has a different priority based on the basic mathematic priorities. Furthermore, several methods in each of the provided classes were not complete, so I completed those as best I could. One feature in particular is not implemented. Any time a negative number appears which is not subtraction, the program does not compute the correct answer. For example, $1--1$ will not result in 2, it will result in 0.

2 Development Environment

- a. Version of Java used: 12.0.2
- b. IDE used: IntelliJ IDEA Ultimate 2019.2.1

3 How to Build/Import your Project

To import the project, open IntelliJ and create a new project from existing sources. All fields can be left default on all menus except the source folder selection. Select the calculator folder as the source folder for the project.

4 How to Run your Project

There are two ways to run the project. Through IntelliJ, one can right click either “EvaluatorDriver” or “EvaluatorUI” and click run. If EvaluatorDriver is run, there will be a console in the bottom of IntelliJ. It will prompt for an expression. Simply type the math expression you wish to execute and hit enter. The process will continue until the user clicks the stop button on the bottom left side of IntelliJ. If EvaluatorUI is run, a rectangular window will pop up mimicking a calculator. The user must click the buttons in order to input an expression. After a user enters an expression and hits “=”, the result will be displayed. This result will not go away. If the user wishes to clear the result, click “C”. The result is not cleared so that if the user wishes to use the answer as the base for another expression, they may do so.

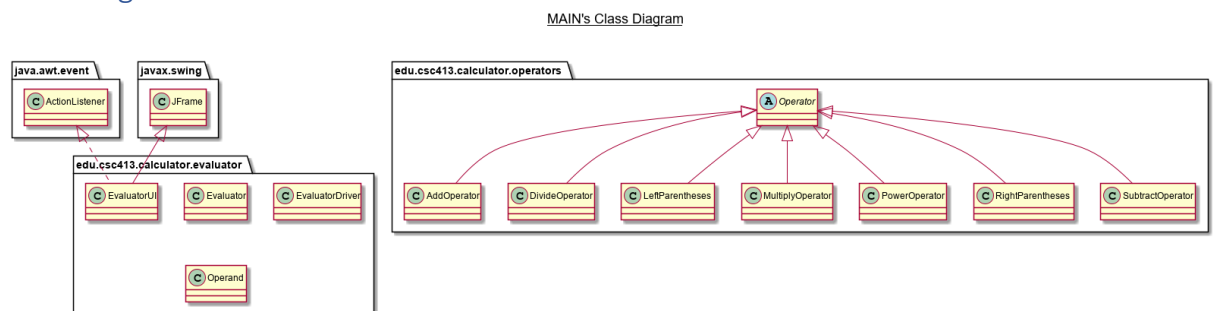
5 Assumption Made

One assumption made is that the operators are limited to only +, -, /, *, and ^. Any other mathematical operators are not allowed. Furthermore, as explained earlier, negative values can't be used in most situations.

6 Implementation Discussion

- a. By using a hashmap, the only entry point into the data is through the execute operator, which utilizes polymorphism and it cannot be immediately clear which operator will be executed.

6.1 Class Diagram



PlantUML diagram generated by Sketchit! (<https://bitbucket.org/jmesmeur/sketchit>)
For more information about this tool, please contact philippe.mesmeur@gmail.com

All of the operators are extended from the Operator class. EvaluatorUI utilizes methods from both ActionListener and JFrame.

7 Project Reflection

In doing this project, I realize that my Java skills were not good enough. I did not know names of methods within predefined classes like stacks or hashmaps. I need to learn more of these.

8 Project Conclusion/Results

In conclusion, this project turned out to be more difficult than I originally anticipated. Handling all of the various situations with different operators proved to be a difficult logical task. I believe this calculator will be able to function properly if a little more time was invested in order to correctly operate with negative numbers in certain situations. However, if using mostly positive numbers, the correct result should be obtained.