

CSC 413 Project Documentation

Spring 2020

Marvin Thai

917432352

CSC 413.02

<https://github.com/csc413-02-spring2020/csc413-p1-mathai1.git>

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	4

1 Introduction

1.1 Project Overview

In this assignment we were tasked to create a way for the system to calculate an equation correctly in the correct order as you would do in math equations. In order to do so you need to first determine the priority in which each operator goes using PEMDAS. PEMDAS is commonly used to determine which operator goes first as it represents Parenthesis, Exponent, Multiplication, Division, Addition, and Subtraction. Once you understand this you would know that Parenthesis has the most priority, then exponent, multiplication or division, then addition or subtraction.

1.2 Technical Overview

In order to implement the methods that will be used it must be understood that the operators and the numbers are separated into two separate stacks when it is detected. Implementing the operator methods are simple as you would take two numbers and call an operator function depending on which operator is in the stack either multiplying, dividing, adding, subtracting, or exponential. The parenthesis are left alone with no methods as it is only used to determine the priority of an equation. Once all of these are determined, combine it all into a function keeping in mind which has a bigger priority.

1.3 Summary of Work Completed

What I have contributed when trying to complete this assignment is creating the operator methods like when two numbers and a operator is given what is done with it. The thing that is done with it is that depending on the operator it would add, subtract, multiply, divide, or exponent the two numbers in order. Once that has been done, since there are longer equations that combine two or more of these operators, I needed to create a way that it would go through the equation and recognize if the value is a number or a operator. When it is a number it would be put onto the top of a stack. When it is an open parenthesis it is automatically put into a stack. If it is an operator then it is put into a stack if the stack is empty or the priority is greater than the operator that is already in the stack. If it is less than then it would remove the operator in the stack and 2 numbers in the number stack and evaluate them and put it back into the number stack. If it is a close parenthesis then it would go backwards until the open parenthesis and evaluate the expression like said before. When the equations have no more values to detect then it would go through the operator stack and evaluate like said before until the operator stack is empty. Then it would finally remove the last value in the number stack and return it.

2 Development Environment

To develop this code I used IntelliJ an Java Development IDE version 11.0.5.

3 How to Build/Import your Project

Since a partially completed file was given to us via github, within github I cloned the file onto my desktop. When it is successfully cloned, on my IDE I clicked File->New->Project From Existing Sources and go into the cloned file and select the calculator file as it contains the codes needed. Once it is loaded in make sure all the files are selected when asked and it should be loaded.

4 How to Run your Project

To just run the program to check if all the function are running correctly, click run all in the test . If you want to run the UI then right click on evaluatorUI and click run, a window looking like a calculator application should pop up and it should be working like a simple calculator.

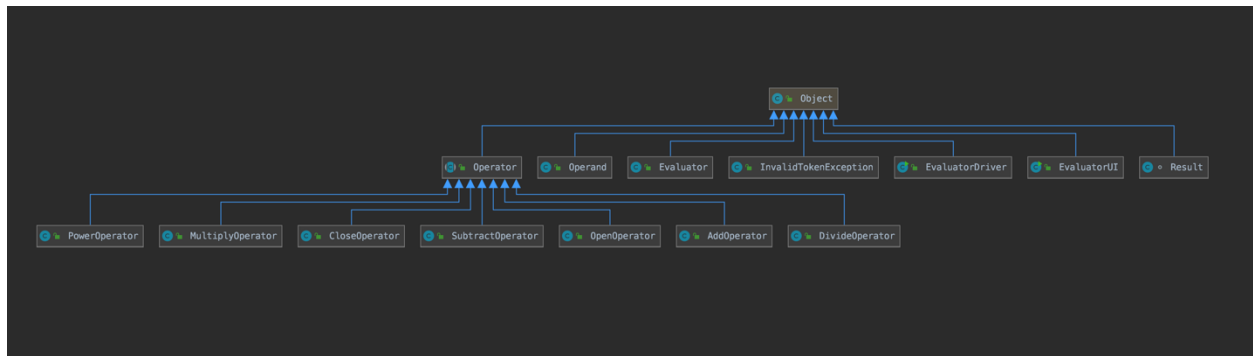
5 Assumption Made

Some assumption that I had made when implementing this program for this assignment, I just assumed all values are returned as integer even though some expressions might not return a integer like how 3 divided by 2 would return 1.5 but in my project since I assumed that It would return an integer only, it would only return 1 leaving off the decimal of .5 since it isn't considered to be a part of an integer.

6 Implementation Discussion

6.1 Class Diagram

Since most of the methods and classes have already been given to us, all we had to do was to create methods for the operators whether its is addition, subtraction, multiplication, division, or power. These operator function are all linked to the Operator class since that is where all the operators are being called upon.



7 Project Reflection

This project that was assigned tested us on the different ways that objects are called. This assignment was not as hard as I had originally thought it would be before fully reading the assignment. Once I took a careful look at what was given and what was required in the assignment it was really straightforward. This project really taught me how to carefully look at what each method would do as it could be vital if a value is not treated properly.

8 Project Conclusion/Results

All in all I thought that this project was really straightforward we only had to implement the operator methods, how each equation was evaluated, and how the GUI would work when the user does something to it. As I have never worked with the GUI before I found this assignment to be very interesting and hope to learn more on the usage of this in the future.