CSC 413 Project Documentation

Spring 2019

Andre Flores

918175055

CSC413[02] Fall

git@github.com:csc413-SFSU-Souza/csc413-p1-good-day12.git

https://github.com/csc413-SFSU-Souza/csc413-p1-good-day12

Table of Contents

1 Introduction

1.1 Project Overview

This project is a calculator that is designed to solve complex mathematical equations and allows us to use parenthesis in our equations as well. While I was given a mostly finished calculator with the looks of everything, I did have to do some considerable work behind the scenes so that the calculator knew what to do when certain buttons are pushed and so it could understand how to solve the equation when asked. This project was a great review for material we went over in past classes and was a good refresher for the beginning of the class.

1.2 Technical Overview

This project is a calculator that uses our own operand and operator classes for the logic behind the calculator. The operator class has several subclasses that include all the common operators used for math like addition, multiplication, etc. These classes were all used along with two stacks, one for operands and the other for operators, to store the numbers and solve the equation given. This project also has a GUI calculator that the user can then input an equation using the buttons provided. After the user inputs an equation, the project then takes that string that was inputted using the buttons, and uses our evaluate function on this string to solve the equation and return the correct answer on the display.

1.3 Summary of Work Completed

For the work I completed I created the Operator classes: AddOperator, SubtractOperator, DivideOperator, ParenthesisOperator, MultiplyOperator, and PowerOperator. I also added to the logic the professor gave us in the evaluate function so that the function can fully solve equations including logic for handling parenthesis. We used two stacks to keep track of the operands and operators respectively and used those stacks to compute the equation. After that I also added to the Calculator UI to add functionality to the buttons and logic so that when the equation was entered a correct solution came back out on the display of the calculator.

2 Development Environment

The version of Java I used was Oracle OpenJDK version 17.0.2. The IDE I used was IntelliJ.

3 How to Build/Import your Project

Copy the repository from Github, then import to IntelliJ using the top calculator folder as the root folder. Then open it in IntelliJ and build the project.

4 How to Run your Project

To run the calculator UI simply go to the EvaluatorUI main class and hit the play button to run the calculator.

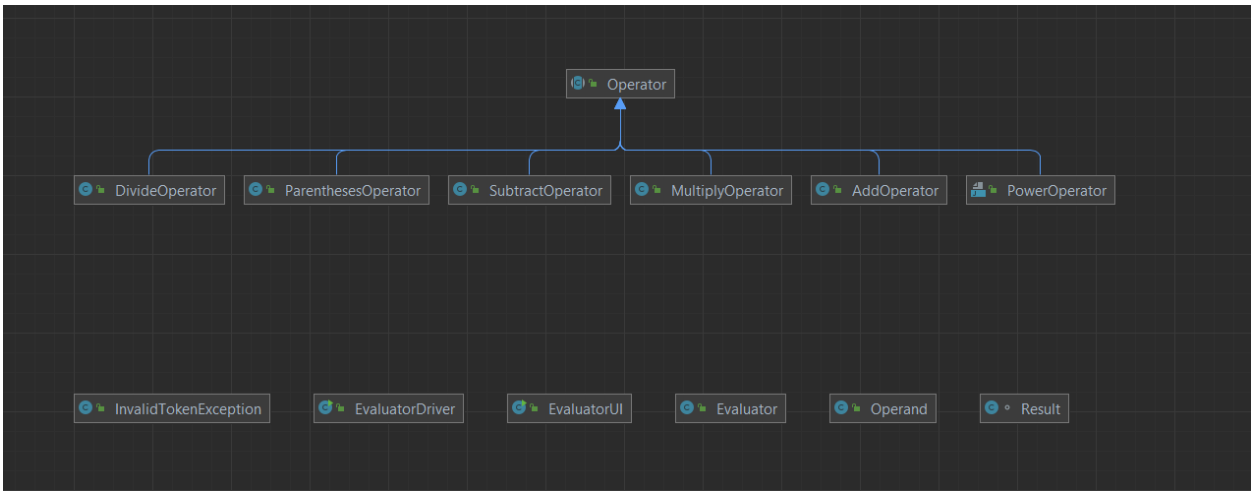
5 Assumption Made

The assumptions I made was that the numbers inputted would all be non-negative integers, however we can output negative answers. I also assumed that the operators would all be binary,

and that the parenthesis would not be used like how they are in traditional math and would not imply multiplication and would simply be used to group numbers and equations together.

6 Implementation Discussion

6.1 Class Diagram



For the Evaluator function, I decided to make one extra parentheses class so I could handle the logic in the evaluator function on how to solve the problem of what the function should do when it encounters a parenthesis in the equation. I used this class so, when our equation first runs into a parenthesis, I could have put a parenthesis object in the operator stack and store it there and continue with the string. Then in the function I had separate logic for when we ran into a closed parenthesis that would solve the equation up until the parenthesis previously inserted in the stack was encountered, then we would store the result in operand and pop the parenthesis object from the operator stack and continue as normal for the rest of the equation.

Then for the EvaluatorUI I used a switch statement to compare the command received from the actionEventObject to strings that the calculator could send. I set up special cases for the “C”, “CE”, and “=” buttons to perform the logic of clearing one character, clearing the entire equation, and solving the equation respectively. Then as a default I added the text inputted to the equation string shown in the calculator. I knew I could do this as a default safely because the UI only allows the user to interact with the buttons provided, so I know they can’t give us any other inputs that would do anything different than the cases I have already laid out.

7 Project Reflection

To be completely honest when I first was saw the code for this project in class, I did feel very overwhelmed and was extremely nervous to start this project. That could also just be the first week jitters going along with that, but I was not excited for this assignment. Thankfully though, that initial fear motivated me to get a real early start on this project and start working on it right away so I could try and give me enough time for any roadblocks I may run into. After doing a more thorough analysis of the code while looking at the assignment requirements and what was expected of me to complete on this assignment, I felt much better about the whole assignment. Once I realized exactly what I needed to do first and what steps needed to be taken, this whole

project felt very manageable. After I started working on it the coding started to come back to me and it was a great review project to start the term. Overall, I'm happy with my work and I hope I can keep that feeling for the next assignments throughout the term.

8 Project Conclusion/Results

I finished the project, and it passed all the tests given by the instructor for all the classes and the overall evaluator function test as well. I also ran the EvaluatorDriver class with auto as the argument in the configurations and it passed all those tests as well so I'm confident the Evaluator function processes the equations correctly. As for the EvaluatorUI, I ran the GUI application several times and had some friends test it as well and every time it gave a correct response to the equation inputted by the user. I am very confident that overall, the calculator GUI works as expected within the bounds of this project.