CSC 413 Project Documentation

Spring 2019

Yakoub Alkabsh

917108002

413 Summer 2023 R4

<https://github.com/csc413-SFSU-Souza/csc413-p1-Yakkubs>

Table of Contents

[1 Introduction 3](#_Toc522827688)

[1.1 Project Overview 3](#_Toc522827689)

[1.2 Technical Overview 3](#_Toc522827690)

[1.3 Summary of Work Completed 3](#_Toc522827691)

[2 Development Environment 3](#_Toc522827692)

[3 How to Build/Import your Project 3](#_Toc522827693)

[4 How to Run your Project 3](#_Toc522827694)

[5 Assumption Made 3](#_Toc522827695)

[6 Implementation Discussion 3](#_Toc522827696)

[6.1 Class Diagram 3](#_Toc522827697)

[7 Project Reflection 3](#_Toc522827698)

[8 Project Conclusion/Results 3](#_Toc522827699)

# Introduction

## Project Overview

* This is a calculator app that lets you do calculations of non-negative inputted numbers.

## Technical Overview

* This is a java program that reads and converts a string into tokens and places them in 2 stacks of either operators or operands so we can evaluate them as a full expression. Operands being a class that holds any valid positive whole number and Operators being class objects that hold an operators priority and its evaluation method. The program then goes through a loop that processes the tokens by popping the operator and 2 operands out of the stacks and returns that value back into the stack until we picked up all the tokens. Then we process the remaining expression.

## Summary of Work Completed

* To get the skeleton code given to us running properly, I need to add 2 more operator classes into the HashMap. LeftParenthesis class that has a priority of 0 and method that does nothing and RightParenthesis class that has the priority of 4 which also has a method that does nothing. The reason for this is because I want the left parenthesis to be ignored until we do the other evaluations and the right parenthesis to be the first thing that is prioritized so we can do its inner evaluation first. With these changes after updating the loop conditions to work with how I have my priority set up, we are able to do the evaluation up until all the tokens are pickup then all we needed to do was finish the rest of the evaluation with a while loop that process the rest of the evaluation until the operator stack is empty. Finally after getting the evaluator class properly running, I updated the evaluatorUI so that it displays the proper text when buttons are pressed and evaluates and clears properly when certain buttons are pressed.

# Development Environment

* IntelliJ IDEA 2023.1.2
* Java 18

# How to Build/Import your Project

* Click on the repo link provided above and click download zip, alternatively assuming you have git installed, you can go to your terminal and clone the repo with git clone <https://github.com/csc413-SFSU-Souza/csc413-p1-Yakkubs>.

# How to Run your Project

* Once downloaded, open folder in intellji and head to the evaluatorUI file folder from calculator > src > main > java > edu > csc413 > calculator > evaluator > EvaluatorUI. and run that file by clicking the run/play button on the top right.

# Assumption Made

* Negative numbers, floats, and letters ideally are not inputted so we don’t need error handle those inputs. We also only do binary operators evaluate binary operators.

# Implementation Discussion

* Created a HashMap that held all the operator types we use so we didn’t have to create multiple if statements for each type of operator, added 2 operator classes with priority 4 and 0 so my while loop could work properly.

## Class Diagram

* Image too big, check Documentation folder

# Project Reflection

* This project took me longer than I expected, My main language is C++ and I only ever took 210 for java so I wasn’t used to using classes like this. Initially I was confused but after watching the lecture again, everything started to make more sense, I think my implementation of the parenthesis classes isn’t the most efficient but it still does the job so I believe I did fine.

# Project Conclusion/Results

* The project works as intended and everything works properly so I feel like I did a decent job with it, the only thing I would change is how I did my loop conditions and priority on the right parenthesis class but all around it still functions properly without too much in efficient code.