CSC 413 Project Documentation

Spring 2019

Carlos Lopez

918559153

413.02

https://github.com/csc413-02-spring2019/csc413-p1-carlosmcodes

Table of Contents

1 Introduction

1.1 Project Overview

The concept that was to be gained in this project was to practice OOP which in how I see it, is making things more general, more portable and more conventional. Making use of classes efficiently and effectively. This was implemented in my program by making characters classes/subclasses versus just making them literal strings.

1.2 Technical Overview

When it comes to technicality, I had a graphical user interface which used several other classes. Those other classes being abstract and concrete. My objective in this was to create objects with each concrete class .

1.3 Summary of Work Completed

I implemented several different concrete classes to compute simple algebra. Aside from this I worked on a function named 'eval' and in this function is where PEMDAS was implemente. Lastly I worked on the GUI to allow for buttons to be pushed followed by the class methods to be executed per operator

2 Development Environment

Java Version:8.0_171

IDE used: Intellij IDEA 2018.3

3 How to Build/Import your Project

Assuming we have git but have not cloned the project, the steps are as follows:

- git clone <project clone with HTTPS>
- go to Intellij, import project from the same directory which it was cloned to, BUT.
- You want to make sure when doing this you find 'calculator folder' then click the 'OK' button then. If you do not do this, then the test will not run correctly and you will have issues.
- Import is complete

4 How to Run your Project

To run, assuming you have the configurations(which test to run) done.

- Click on run on the top bar
- You choose the EvaulatorUI test because this test uses all classes therefore checks all issues.
- Calculator pops up, program is up and running.

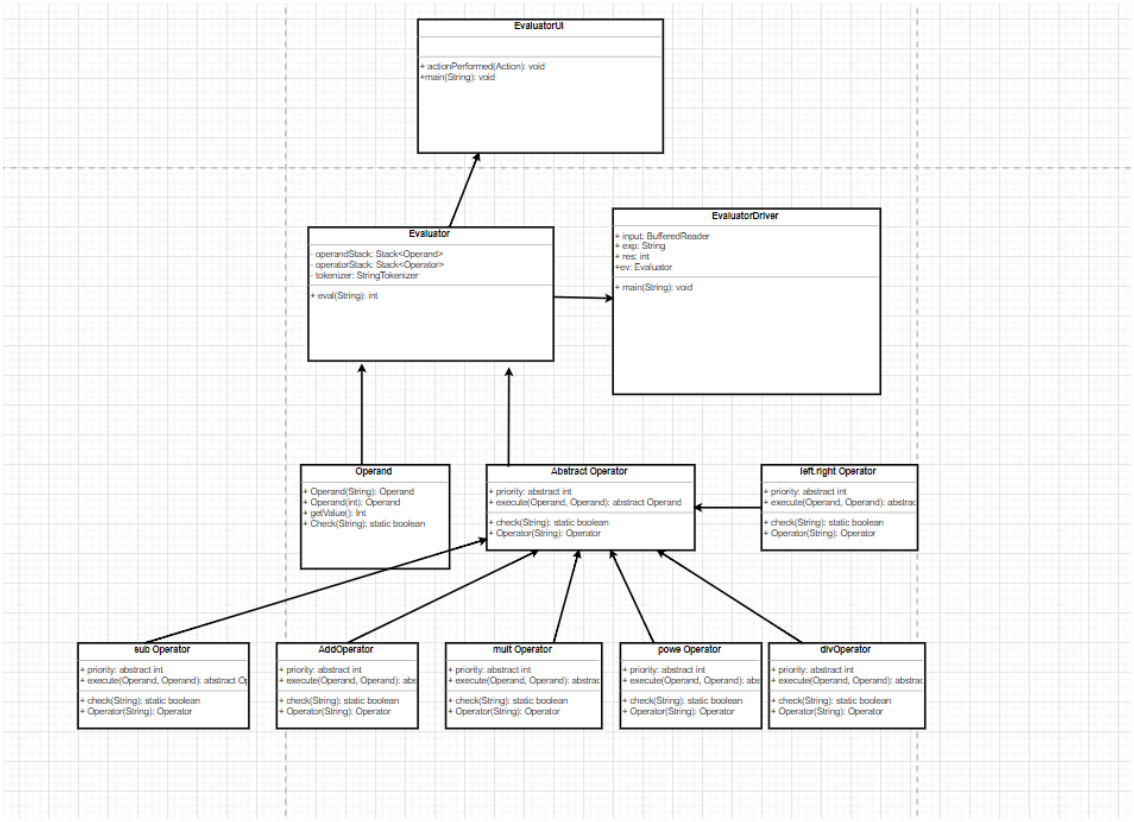
5 Assumption Made

My assumption was that I would not be able to finish the project. Though I did complete about 95%, my assumption relating to the actual project was that it would be a lot harder then expected. I assumed making the sub classes would be difficult but they were not as such.

6 Implementation Discussion

Before I start any project with multi classes I always like to do a UML design just to give myself a better visual as to what goes where, who knows who and what uses what in which ever way. Below I have a picture of my design. Though the arrows are all wrong, it still gives me a better visual of which class is interacting with which.

6.1 Class Diagram



7 Project Reflection

To reflect how I did and on my algo in eval, I was not too happy with it. I see a lot of flaws in it, though I do not have enough time to tinker with it. I wish I had less conditionals. In reference to the sub classes and abstaction, I feel I did well. I thought I would have a harder time with this issue but I did not.

8 Project Conclusion/Results

To conclude, I feel I did sufficient. Though I did not 'pass' all the test in the tester. In relation to emphasizing OOP, I feel I did good. I struggled a little understanding how different classes were used but the graph really helped me get a visual on it. In result the calculator works great. PEMDAS is done as expected and the calculator does not crash. Therefore I am content with the result of my work.