CSC 413 Project Documentation

Summer 2024

Student name: Guiran Liu

Student ID: 923620812

Class.Section: CSC 413-02
LEC (1677)

GitHub Repository Link:

https://github.com/csc413-SFSU-SU2024/calculator-GuiG2023

user name:GuiG2023

Table of Contents

1 Introduction

1.1 Project Overview

This is a calculator project, which means we are going to build an “over-engineered” calculator. It is made of two parts: the first part is the graphical user interface, and the second part is about how to calculate the input number using some algorithm and data structure. When we put the two parts together, it becomes a small project, and it should be a calculator that allows the user to evaluate integers with particular operators.

1.2 Technical Overview

The project and codes are not complex. The professor has already given us the skeleton and code, which includes two different folders: one is an evaluator, and one is the operator; we need to design our own calculator classes and pass all the technically tested tests in my calculator, especially the evaluator folder. I have four classes: evaluator UI if valid token exception and operand classes. In my operator folders, I have different classes. First, I have an operator abstract class, meaning it is the operators' parent class. All this structure is about oriented-object programming I'm trying to connect all the classes and improve the efficiency of the project considering runtime complexity and space complexity

1.3 Summary of Work Completed

Completed：

I completed all the requirements from project one, and I also pass through all the tests. Most specifically, I implemented and created different classes in the operator folder extending from abstract operator class, and also I implemented the algorithm in the evaluator class, which is the most important part of the calculator considering how to use stack push and pop to deal with operands and operators, after finish all those things I learn and finished the graphic user interface part and make it becomes a real calculator that could be used by a user.

Requirement 1:Implement the Operand class.

Requirement 2: Implement the Operator Abstraction and its sub-types. See the starter code for information.

Requirement 3:Complete the Implementation of the above algorithm within the Evaluator class.

Requirement 4:Reuse your Evaluator implementation in the provided GUI Calculator (EvaluatorUI.java).

Requirement 5: (Design Restrictions)Please make sure that class members (static or not) have the correct access modifiers. You will be graded on correctly using the private, public, and protected access modifiers. Classes should not directly access the data fields of another class.

Requirement 6: (Design and Unit Test Requirement)Please use the following names for the operator classes. This will improve the performance of the given unit tests. If not, you must modify them.

2 Development Environment

a.

openjdk 21.0.2 2024-01-16

OpenJDK Runtime Environment (build 21.0.2+13-58)

OpenJDK 64-Bit Server VM (build 21.0.2+13-58, mixed mode, sharing)

b.

IDE: IntelliJ IDEA 2023.3.3 (Ultimate Edition)

3 How to Build/Import your Project

I cloned the GitHub repo from the professor to my own local repo, then I used intellij open to find the repo trust the project and add it into my working environment, which is my intelligent idea.

4 How to Run your Project

There are different ways to run my project. The easiest way is to use a terminal. We can copy and paste or simply type the command the professor gave us. This way, we can compile the project and try to run it. The other way is to run the project in IDE to find the main method of the graphical user interface class, click the triangle arrow, and then the program will run.

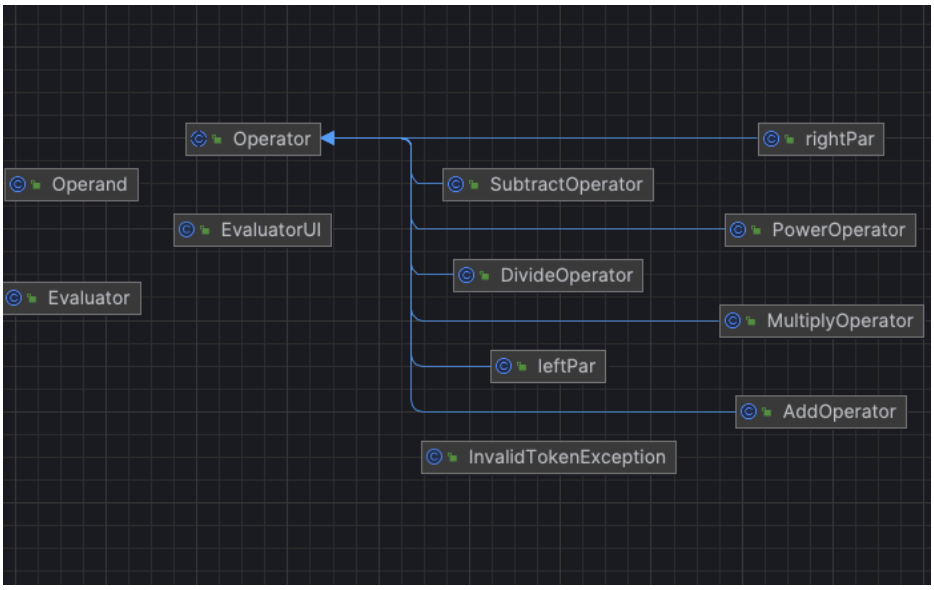
5 Assumption Made

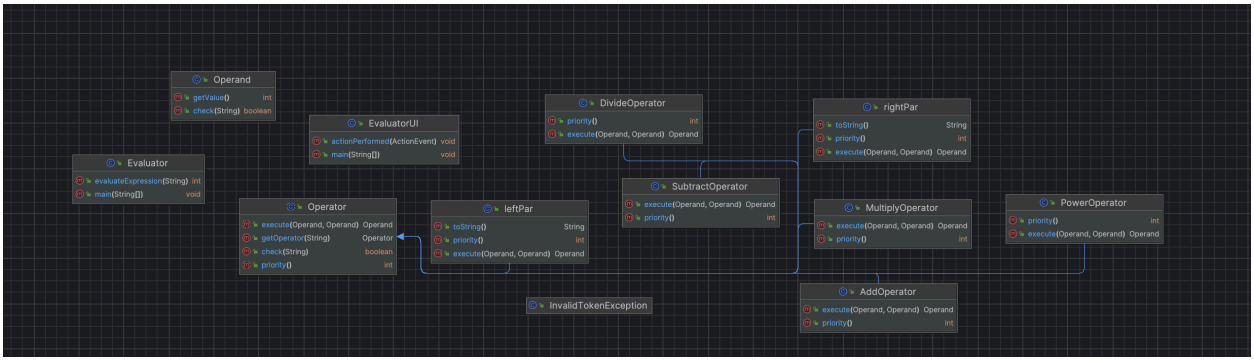
The assumptions are we assume that there are no float numbers, only integers, we also assume that we have no negative numbers, and we assume that there is no complicated math calculation; technically, we assume that it's a very simple integer calculator.

6 Implementation Discussion

According to the professor's lecture and what I did from asmt, I found that the organization of my code and efficiency is really important, so that's why I followed the professor's instructions using the abstract class operator class and creating different operator classes inheritance and implement different functions in it. Also, we can always so sad the original operator class as an interface and also implement the interface to different kind of operators is a different ways to implement and also in operator class we are using hash map to create a key value connected container and using static curly braces to initialize different kind of operators including add subtract multiply power divide even left. Right parentheses in this way, we can only initialize it once, which means we don't have to create it again and again when we create different operators. It's also improved this space usability.

6.1 Class Diagram





7 Project Reflection

When I do the project, I met a lot of problems issues but finally I fixed most of them but the first time I got my project and skeleton code I even don't know where to start then I follow his professors instruction start from the operand class because operant class is a kind of individual class it doesn't depends on other classes, then I go through all the classes professor gave us I try to figure out how these classes could make my project into a calculator after several hours I I fully understand and try to refresh my knowledge about data structure and algorithm I started to know what I'm going to do and then I use the operator abstract class to create different operators by using different operators just push and pop them into the operator stack and we also have a all brand stack which is using to push and pop operands so by using these two different stacks I know how to control the expression how to get the correct result. But when I try to implement my algorithm into code I failed several times the first time I fell because I cannot pass the test I don't know what's going on but just failed past I use the debug tools by intellij try to find things line by line finally find that in one of my parenthesis class I forgot the two string method which means it might evaluate a class I can not recognize which one is parenthesis, after that I fixed it and pass all the tests.

8 Project Conclusion/Results

My result is that I finally passed out a test but I still want to say if professor let me start from zero from scratch to build a calculator I canard no all the steps and fully completed like this because I think I still need to learn the high level structure of software development and I know I need to practice more design patterns.