CSC 413 Project Documentation

Spring 2019

Dinesh Thapa

920879242

CSC413-02

https://github.com/csc413-sp21/csc413-p1-dthapa770

Table of Contents

1 Introduction

a. Project Overview

The following project demonstrates the function of a simple calculator, calculates infix expression which is generally complex than other types of expression for computers due to precedence level of operators and displays the results.

b. Technical Overview

This calculator can perform the calculation with the arithmetic expression: + , - ,* ,^ , / and when the user gives input expression, operand and operator stack keeps track of our operand and operator from the expression.

c. Summary of work completed

Most parts of the program were already given to us and my contribution to the program is Operator Class, Operand Classes, Evaluator class, and EvaluatorUI, to add action to the button.

2 Development Environment

a. Java Version: 11.0.9
b. IDE: IntelliJ 2020

3 How to Build/Import your Project

1. Clone or Download Repository from GitHub: https://github.com/csc413-sp21/csc413-p1-dthapa770.
2. In IDE, import the downloaded project and select calculator as the source root of the project.
3. Create project from existing resources
4. Import all the libraries prompted by the IDE to run units test, as well as modules and JDK, is required.
5. After importing is finished, build the project.

4 How to Run your Project

1. Go to: Calculator -> out ->production -> main -> edu -> csc413 ->calculator ->evaluator
2. Run: EvaluatorUI.class

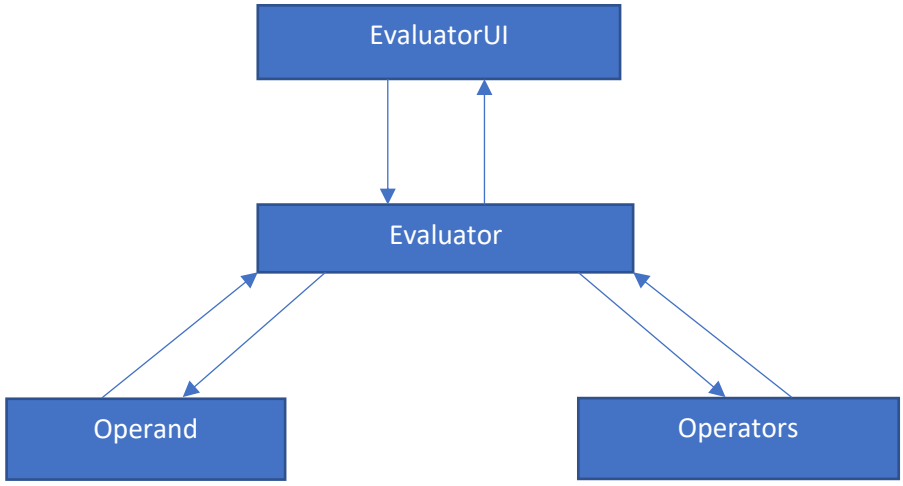
5 Assumption Made

1. We assume that the calculator can only accept Integers and operators. Integers include: 1,2,3,4,5,6,7,8, 9, 0 and operands includes: +, *, /,-,(,),^.
2. Grouping is done for the operands by priority, higher priority operators will be evaluated first.
3. Parenthesis cannot do multiplication functions.
4. Running test successfully for all the cases provided ensures our program is running as it should be.

6 Implementation Discussion

a. Inheritance
b. Encapsulation
c. Polymorphism

6.1 Class Diagram



7 Project Reflection

The Project was at least for me was a challenge, especially implementing the brackets, handling those brackets took almost 5 hours to figure out the correct and proper implementation. Luckily, most of the programming was already done, and adding the remaining portion of the code was easy but figuring out the structure and getting a general idea of classes and their working relationship was time-consuming and I think it's a crucial part of any programming to analyze first. After hours and hours of debugging Evaluator class, passing all the tests surely was a relief.

8 Project Conclusion/Results

After completion of this project, my knowledge of OOP concepts got enhanced and I have surely become a good debugger. The correct implementation of abstract classes, inheritance, and polymorphism gave me a good foundation of OOP. I wish we could have some more time to improve the existing program.

To improve this Calculator, the functionality of Parenthesis can be implemented to calculate expression inside it and it can be developed further to upgrade it as a scientific calculator which could solve the complex mathematical problem.