CSC 413 Project Documentation

Spring 2019

Bahar Moattar

Student ID:

920006396

Class.Section:

413.02-Spring2019

GitHub Repository Link:

https://github.com/csc413-02-spring2019/csc413-p1-baharmoattar.git

Table of Contents

1 Introduction

1.1 Project Overview

The Evaluator project with basic operations provides user various mathematical operations. The base class in the application has all the methods for calculation. Based on user input, it calls the corresponding class and the user can get the respond as an answer to different simple mathematical equations. It also provides a message when wrong input or choice is given.

1.2 Technical Overview

The Calculator project with the evaluation helps to learn how to create a Java console application, use different Java features like inheriting, looping, packages and I/O classes, and conversions in different types. This project helps to learn how to make a border in different sizes, add objects to it, request an input in this Java built area, validate the request, process and provide the output.

1.3 Summary of Work Completed

To create Calculator, 5 main java files were created. The base class “EvaluatorUI” is the one that’s make the calculator and prints the input and output and has all the methods for calculation in it. one main class “EvaluatorDriver” for showing all the work, and run different kinds of tests to get the right answer. “Evaluator” class is to make all the different conditions work. Like when we have parenthesis or click on different delimiters. “Operands” to let the output gets calculated correctly by checking different conditions, and “Operator” is to assign the way each Operand needs to be calculated. Other classes for each

operator made to give the instruction of action. All will be running through some tests to make sure its functioning correctly.

2 Development Environment

a. Version of Java Used:

   1.8.0-191-b12

b. IDE Used:

   IntelliJ

3 How to Build/Import your Project

this project has to be uploaded form the GitHub. There is a link in iLearn provided by instructor to direct it to the GitHub and my personal private depository.

1. Click on that link and go to the assignment page in GitHub.
2. Click on “Clone or Download” and copy the path.
3. In terminal environment login to your git account and type “git clone “the path address you copied”
4. This will make a version of my project in my computer and terminal to have access to.
5. Open the IDE (for me it was IntelliJ). And click on “Import Project”.
6. The window will open to brows in your computer to find the project you just made through the git clone in terminal.
7. Click on that and “open”.

8. In the next screen pick “Create project from existing sources”, “Next”.
9. Do not change the project name in the next screen and just hit “Next” again.
10. In the next screen make sure the path of this project in your computer is checked and click “Next”.
11. “Finish”
12. When the project is loaded and imported to IntelliJ, you will see the whole package and classes that’s been transferred to it.
13. Each time you make some changes in your program through this IDE, open your terminal environment, type “git add . -> git commit -m “some message” -> git push” so all the changes have been made to the program will be saved in GitHub.

4 How to Run your Project

As stated before, there are 5 different classes in this project and I made 7 more to add to the operator signs and their functionality. For each class, we can run the project by clicking on the play button. Although the program may run without any bug in that case, but there are 8 different classes as a test to run. Meaning that the program needs to pass those tests in order to function perfectly. First, we can make sure there Is no error in our first 5 classes. We can run EvaluatorUI and EvaluatorDriver individually to make sure those are running with no errors. For each one of them we click on the play button to see if there is any bug to fix. Then we will move to the test runs. For each test class, we can click on the run button and see if there is any test that fails and needs to be fixed. It usually shows the

line number that the test fails, and we can directly go to that number or the class its referring to fix the issue.

5 Assumption Made

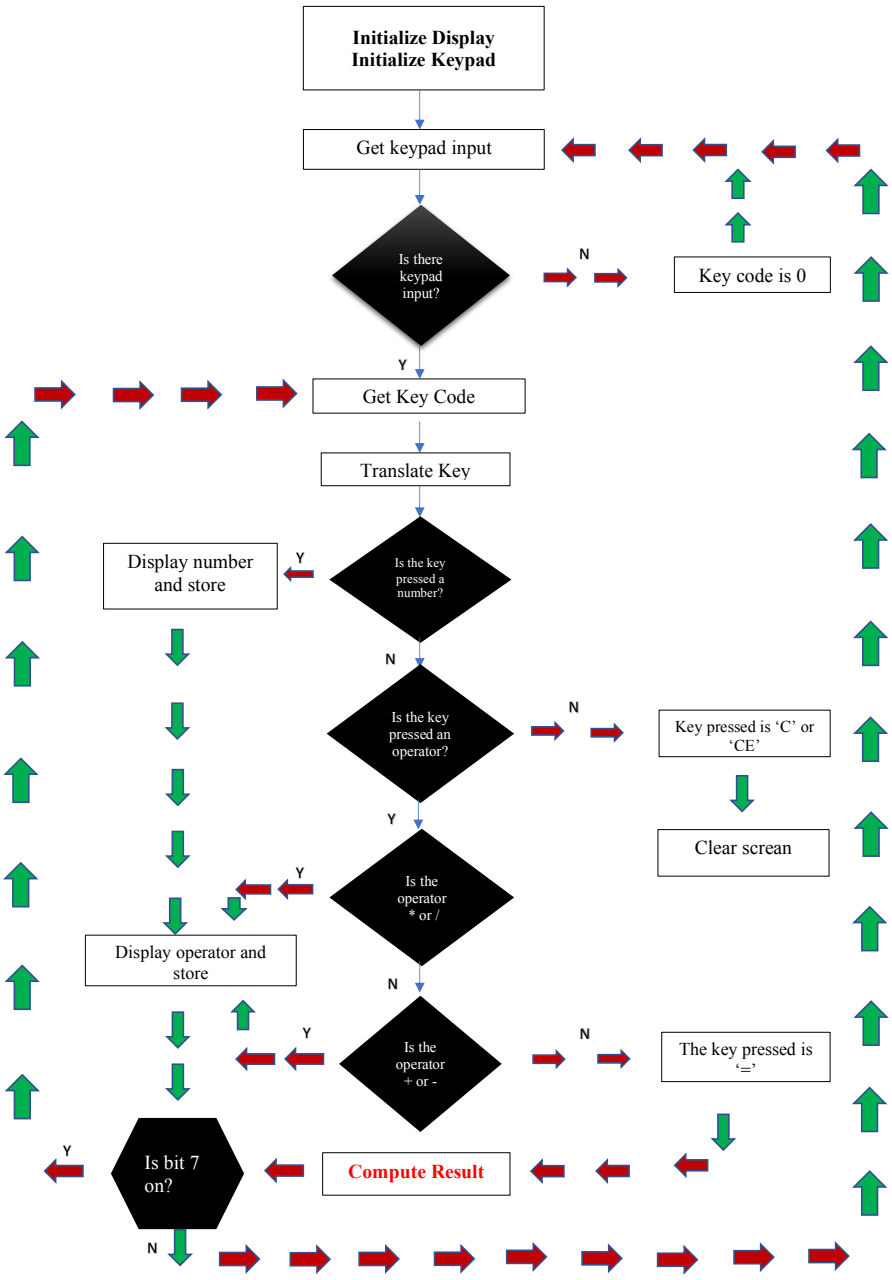
I mostly use the assumption for classes I make. Like the classes we used to complete this project, we mainly used basic operations, names, addition, subtraction, multiplication, division and power. For each one of these signs you have to make sure to add the “first number”, operator, and “second number”. Each one of them needs to stay in the buffer and use to get the result from the equation and then show the result. For each operator, for example addition, I assume that I’m writing this down in my notebook: 2 + 3 = 5, I need to give the first number as a user input (lets say to my brain), operator: this sign of operator needs to be store as addition, so every time in this code we get to this line the system already knows how to translate this operator. Exactly the same as our brain process to get the answer to this equation. In elementary school we learned and saved the meaning of addition sign, and then whenever and wherever we see this sign we already know what it means and what is it supposed to do. Then the second number, and = operator. This operator needs get each input, evaluate them, and give us the result. Same process as our brain. After we saw the addition sign, we add the number together and the result is what we say or we write as an output. This process of assumption is for every operator and sign. With all being said, this project will be implemented based on these assumption:

a. The visual basic programming language will make the program run on any computer
b. The project will ensure accurate processing in computer
c. The facts and figures are reliable, the designed software to run primarily.

6 Implementation Discussion

6.1 Class Diagram

Below is a simple graph showing how to implement a calculator assignment algorithm and get the result:



7 Project Reflection

The lines of codes that inspect other lines to find any error. In this project the EvaluatorDriver Class is the class that's running all the tests to fail or pass them and catch the errors.

8 Project Conclusion/Results

The basic idea of the software solution for a calculator is to parse the string created by keys typed and execute it (when controlling unexpected sequences of keys at the same time). Also the state machine required by any software is a result of analysis and design. First, we need a clear picture of the task we are going to do and then we will start all the condition problems.

This project has established a user-friendly calculator and the impulse response. It is possible to make it easier for the user by adding more variety of functions to it. but in general, if in this project we are able to introduce user defined functions then it will give user what he/she wants, that will be the success of the attempts.