Student Name: Giovani Rosas-Marin

ID: 916106144

Students that worked on ideas:
Diana Yu Yu
Hector Aguirre
Carina Kalaydijan
Derick Smith
Christopher Rosana

CSC 413 - 02

Repo:

https://github.com/csc413-02-su18/csc413-p1-gio

vannirosasmarin.git

1 Introduction

Project Overview

The purpose of the assignment is to practice object oriented design to make a calculator. This will make us able to do two programs that can evaluate mathematical expressions and artificial calculator that is a GUI calculator The Teacher provided the code where some were already completed like EvaluatorTester and some where partially implemented were we had to complete them with research and coding an.We had to do utility classes (operand & operator) being used in Evaluator. The Evaluator class is able to implement a single public method called “eval” and takes a single string parameter that represents an infix mathematical expression. These expressions will be evaluated and will return an integer result.

Technical Overview

-stacks -hashmaps -while -extension**s** -Switch statements -For Loops -If statements

Summary

After completing the assignment with only some minimal problem with the negative sign the process given in class and code was basically the same the only difference is that I started backwards to be able to understand the other code that was not totally explained in class. But just created subclasses for the operator and combined the operand check the excuse token. For the evaluator had to implement a switch statement to do the stack

part of the operator. The GUI was basically given the big hint in class on how to implement the code of the condition of the actionPerformed.

2 Development Environment

- Java Version: Java 8 update 151
- IDE: NeatBeans 8.2 and Intellij 2017
    - NetBeans 8.2 was used to import to project.
    - Intellij 2017 was used to develop the project and debug because netbeans did some weird errors in debugging my switch statements.

4 Project Steps (Build, Import)

In order to build or import the calculator in the IDE follow these steps:

1) At the first page there is a link of the repo. Click on the link.
2) On the right hand side you will see a green tab that says “Clone or download”. By clicking on this it should give you a couple of options whether you wish to clone it will give you the link again that can be used in git bash to clone it there or just downloading the file.
3) In case you downloaded the zip file.
4) Also remember where you downloaded the file in.
5) Now go to your Downloads and locate the zip file you just downloaded.It should have name of : “csc413-p1-’username’-master” .zip
6) Click on it and the file should convert into a folder.
7) Click on the folder and there should appear 5 items.

8) Open the folder that says “calculator”
9) Open the folder evaluator. And there should be java classes on them.
10) Right click on one of java class.
11) Choose “open with” in Netbeans.
12) Repeat steps 10 and 11 until you have opened all the java classes. NOTE: it should be one Project.
13) Now go back to the previous folders where it had 2 other folder named “evaluator” & calculator”
14) Click on the “operators” folder
15) Right click on the java classes as we did from steps 10 and 11.
16) Once the projects have been opened, the program is ready to “run”

The steps above are for people with less knowledge of importing new projects to your pc

If your going to the easy way follow the next steps:

1) Go to the link in the first page
2) When you get to the Github repository there is a green button that has “Clone or download”. It will open small box with two options of download zip or open in desktop that are for the option on top. For this just copy the link provided there that is the same as the repo in the first page.
3) In your pc open Git Bash(if you do not have it yet installed download it first then continue with steps)

4) In git bash type “git clone repo link” for example “git clone https://github.com/csc413-02-su18/csc413-p1-giovannirosasmarin.git” but after you type git clone you should right click and it should paste the link of the repo because you copied it before.
5) After cloning it will be located in the user used when cloned.
6) Open your Intellij and go to “ File”
7) Click on “Open…” it should open a new floating window displaying all the folders located in your Local Disk(C:)
8) Click on “Users” is will open a list of user select the user you used when cloning the repo
9) It will open the user and give you other list of folders. There you will find the folder repo named “csc413-p1-’username’(username is the persons name of the repo).
10) After you select it it will open the content of the repo select “calculator” and click “ok”
11) It should take some seconds to open all the project
12) The next steps are to locate the folder of the repo
13) Open “File Explorer”
14) Click on “This pc” it should take you to the files that are in your pc.
15) Click on “Local Disk (C:)” it will have some folder containing programs and your documents that you have stored.
16) Click on “Users” select user used when cloning

17) Look for the file named “csc413-p1-’username’

18) If open folder it will contain the pdf of the assignment and the folder “Calculator” where all the code is located. In “documentation” will be the report pdf of the assignment.

5 Project Steps (Run)

Once you have imported and able to open repo in the IDE successfully, you have two options. You can run either the EvaluatorUI or the EvaluatorTester.

For Intellij to run the **EvaluatorUI or EvaluatorTester:**

1) Open the the project Calculator and the classes EvaluatorUI and EvaluatorTester
2) At the top of the IDE you will see a toolbar “File, Edit, View,...” look for “run”
3) The bar will give you options. Scroll down onto the one that says “Run ”
4) It will pop a new floating window with option the EvaluatorUI or EvaluatorTester
5) Select first EvaluatorTester to check that the “test” works it will give you the results of the Evaluator.java and the “res” given in the main if EvaluatorTester.
6) And to run the EvaluatorUI repeat step 2 to 4 but now select EvaluatorUI to run

7) The calculator should pop up. Feel free to click and calculate any math problem you desire.

.

6 Assumptions

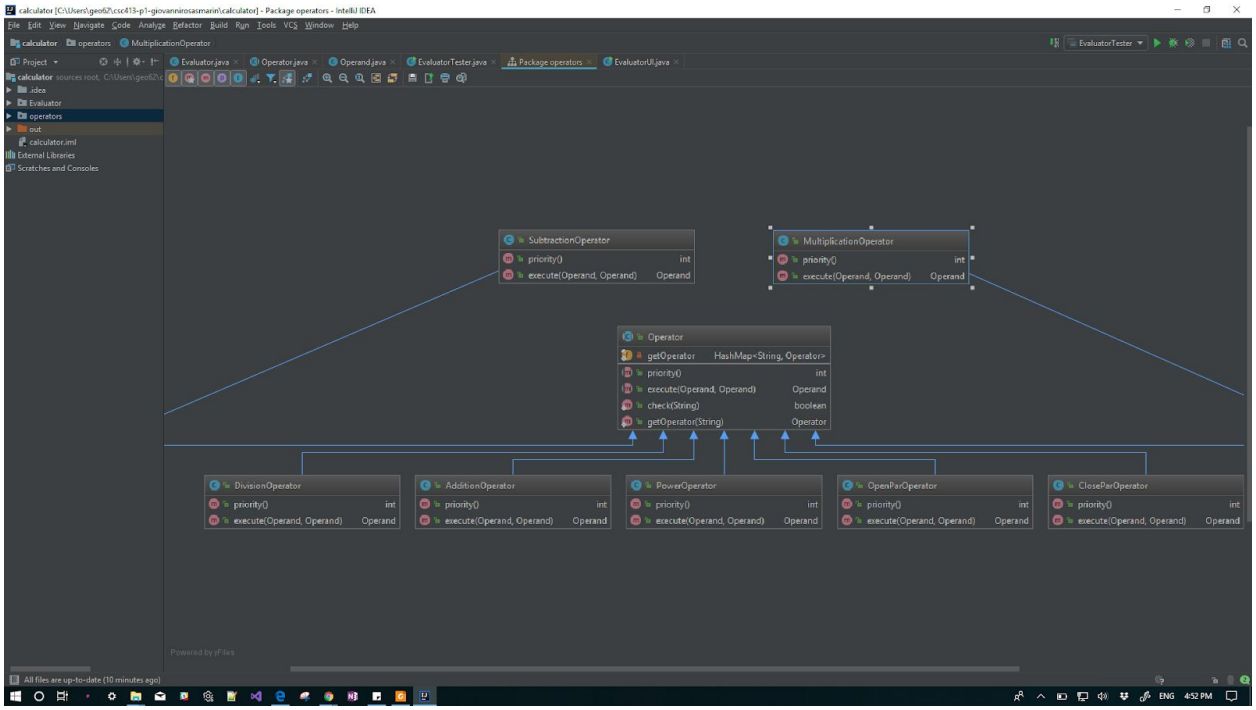
Before cloning the assignment 1 in class Antony Souza that is the teacher for CSC 413 in SFSU explained the requirements and what the calculator would do after completing the code missing in Evaluator.java, EvaluatorUI.java, Operand.java, Operator.java and all the subclasses that where needed. I just thought abouts if statements for all the conditions in EvaluatorUI and while loop missing for the conditions of the tokens in Evaluator and for Operator I had no Idea what to expect to produce the necessary code like the Hashmap because he wanted us to do research on it. But for the subclasses for the operators it was just to state what math procedure each operator had to make to getOperator (answer). After Cloning and seeing the code at home had more worries about not being able to complete the assignment because of the missing knowledge of some code required to make the project function. I started by operand and operator to implement them because they are required to make the other classes work.

7 Implementation Discussion

I started implementing the Operator.java because that's the base of the operation needed for the calculator to work. I made the hashmap based on an example found in stackoverflow where it explained the function of it and what was necessary also the teacher had given a hint in the comments that he provided in the assignment. Created a

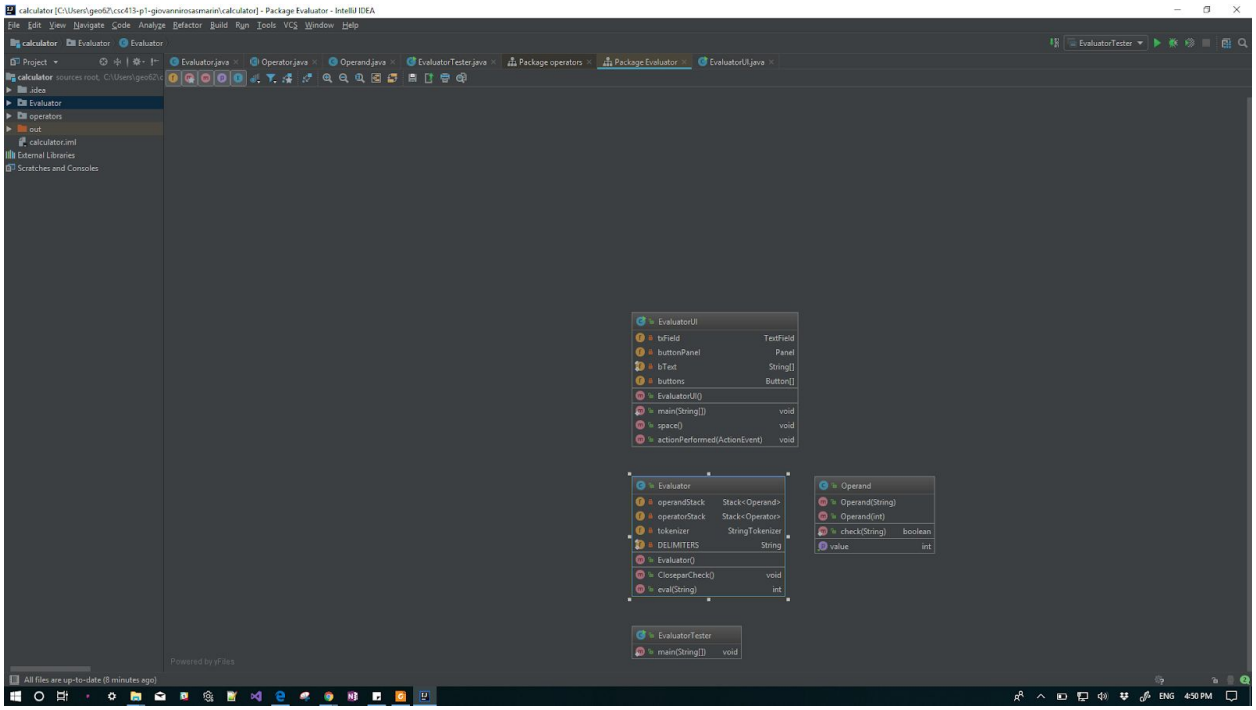
private static hashmap named getOperator to be easy to distinguished when implementing the evaluator.

The static getOperator.pu(“token”, new “operation name”)) ; was the hashmap concept for all the operation need for this calculator



After implementing that I continued to make the check if when the EvaluatorTest run to check for the token “-+*/^()” if any of those tokend where detected if would return check that is true or false because I made it boolean. Then to get to operation of the token is where the hashmap comes in use becuase the token goes in and looks for the subclasses that I made for each operation that is super basic for example the addition was extending Operator and needed to override the default priority and give the operand a thing to execute so it overrides the operand class located in Evaluator package. To make it function needed then to make operand so went to give fill out the

given code. The Operand had to give only integers and return the value. Check if the token was an integer if not it would say false and if it was an integer it would return true. Back the addition the Operand executed to operands op1 and op2 and created a new operand to make the addition of the op1 + op2 and it would return the answer. Then that was done for division, multiplication, power, the parenthesis close and open and subtraction. Just changing the token and priority given in assignment. I moved to Evaluator where my classmates gave me and overview of class material provided by the teacher of what was necessary to do to make Evaluator work to run EvaluatorTester. Started by given condition for the parentheses because the requirement said not to make them push to the stack or execute them. So if it locates”(“ it would go and check the operatorStack then continue after doing to while loop. If “)” was found it would only push “)” ti the stack it would continue to check if the stack is not empty and do the code provided by the teacher. I did some basic switch statements to locate priority and the size of the stack because my logical error before was that was running with one loop before and didn't had it at the end to do the last operation of the operands. Didn't find the logical problem for the negative “-”.



8 Reflection

The reflection would be totally different if this assignment was done in regular semester and not the summer. But for the summer downgrading the harsh requirements for the code would be good because some of us have to work full time to live in the city because I get full scholarship only in the regular semester only. I had less time to do the assignments than other student not saying that they don't have other things to do than school. But I tried my best to accomplish all the requirements of the assignment only have like 2 problems with “-” token and the cleaness of the code I made.

9 Conclusion & Results

Its kind difficult to take classes in the summer because of all the load of work that has to be done in short amount of time and because we have to look up information that is not provided

in the lecture but its also good to do your own research to be able to learn more stuff in your own. I thought that the hashmap was going to be more hard but it ended up being simple to implement in the operator. And all the subclasses done was just remembering some csc 220 info to do the override. The overall result in not what I wanted in the EvaluatorTester because the last problem required gives me -926 and the answer has to 1176 but after debugging it I found the "-" is only recognize for the subtraction but not to make the next number next to it a negative number. I tried making a condition for that problem but got even worse so I left it for a while then started to implement to EvaluatorUI and found out that the negative works correctly there in simple math problems. Just when adding a huge problem is when the code starts to give errors in the switch statement that I made in Evaluator. I feel good about the result I already had low hope of accomplishing all the assignment because I had some errors at first in the logical sequence of my switch statements and also found a space inside the "-" in the bText that made my calculator crash each time that you clicked the negative sign but its all fixed.