Even Semester (2020)



**BINUS UNIVERSITY**

**BINUS INTERNATIONAL**

|  |
| --- |
| **Assignment Cover Letter**  **(Individual Work)** |

**Student Information :**

1. Davin Pratama Chandra – 2301891090

**Course Code :** COMP6510 **Course Name :** Programming Language

**Class :** L2AC **Name of Lecturer :** Jude Joseph Lamug Martinez

**Major :** Computer Science

**Title of Assignment :** Calculator

**Type of Assignment :** Final Project

**Submission Pattern**

**Due Date :** June 20, 2020 **Submission Date :** June 19, 2020

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

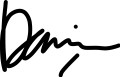
**Plagiarism/Cheating**

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

**Declaration of Originality**

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student :



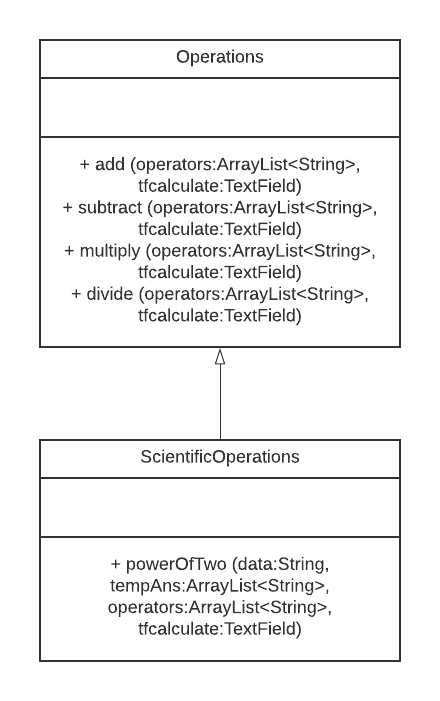
Davin Pratama Chandra

1. Project Specification

This project is a simple calculator GUI (graphical user interface) made using JavaFX on IntelliJ IDEA. This project allows the users to do multiple mathematical operations at once according to the mathematical order of operations.

When the user runs the program, it will display a calculator with a text field and several buttons for number and operation inputs on a GUI. The user can operate the program like calculators in general by pressing the number and operation buttons. Every time the number buttons are pressed, the numbers will appear on the text field. If the ‘clear’ button is pressed, the numbers on the text field will be deleted. When the ‘equals’ button is pressed, the operations that the user has inputted will be solved.

1. Solution Design
2. Class Diagram



These are the classes that I use in the code. ‘Operations’ and ‘ScientificOperations’ are used when the mathematical operation buttons are pressed. ‘ScientificOperations’ class is the inheritance of ‘Operations’ class.

1. Discussion (Explanation of The Code)

***Sample.fxml***

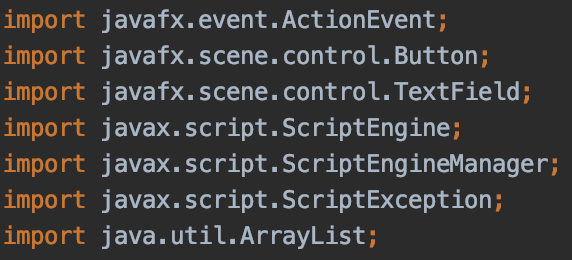
This file is used to create the user interface. It includes all the elements

shown in the user interface, like buttons and text fields. The elements in *sample.fxml* are:

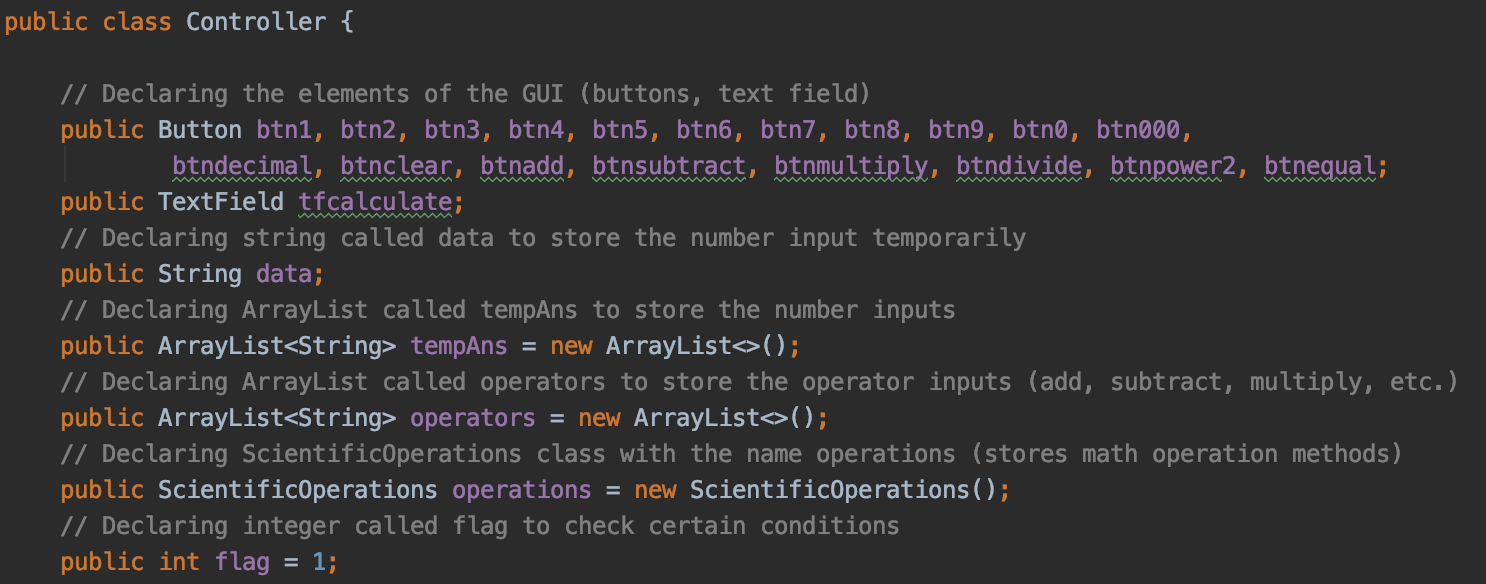
* tfcalculate = text field of the calculator (showing the pressed numbers).
* btn1 = button for number one.
* btn2 = button for number two.
* btn3 = button for number three.
* btn4 = button for number four.
* btn5 = button for number five.
* btn6 = button for number six.
* btn7 = button for number seven.
* btn8 = button for number eight.
* btn9 = button for number nine.
* btn0 = button for number zero.
* btn000 = button for three zero numbers (000).
* btndecimal = button for decimal points.
* btnclear = button to clear the text field.
* btnadd = button to add numbers.
* btnsubtract = button to subtract numbers.
* btnmultiply = button to multiply numbers.
* btndivide = button to divide numbers.
* btnpower2 = button to square numbers.
* btnequal = button to solve the equation.

***Controller.java***

This is the class that controls the function of the buttons, so every time a button is pressed a method will be called.



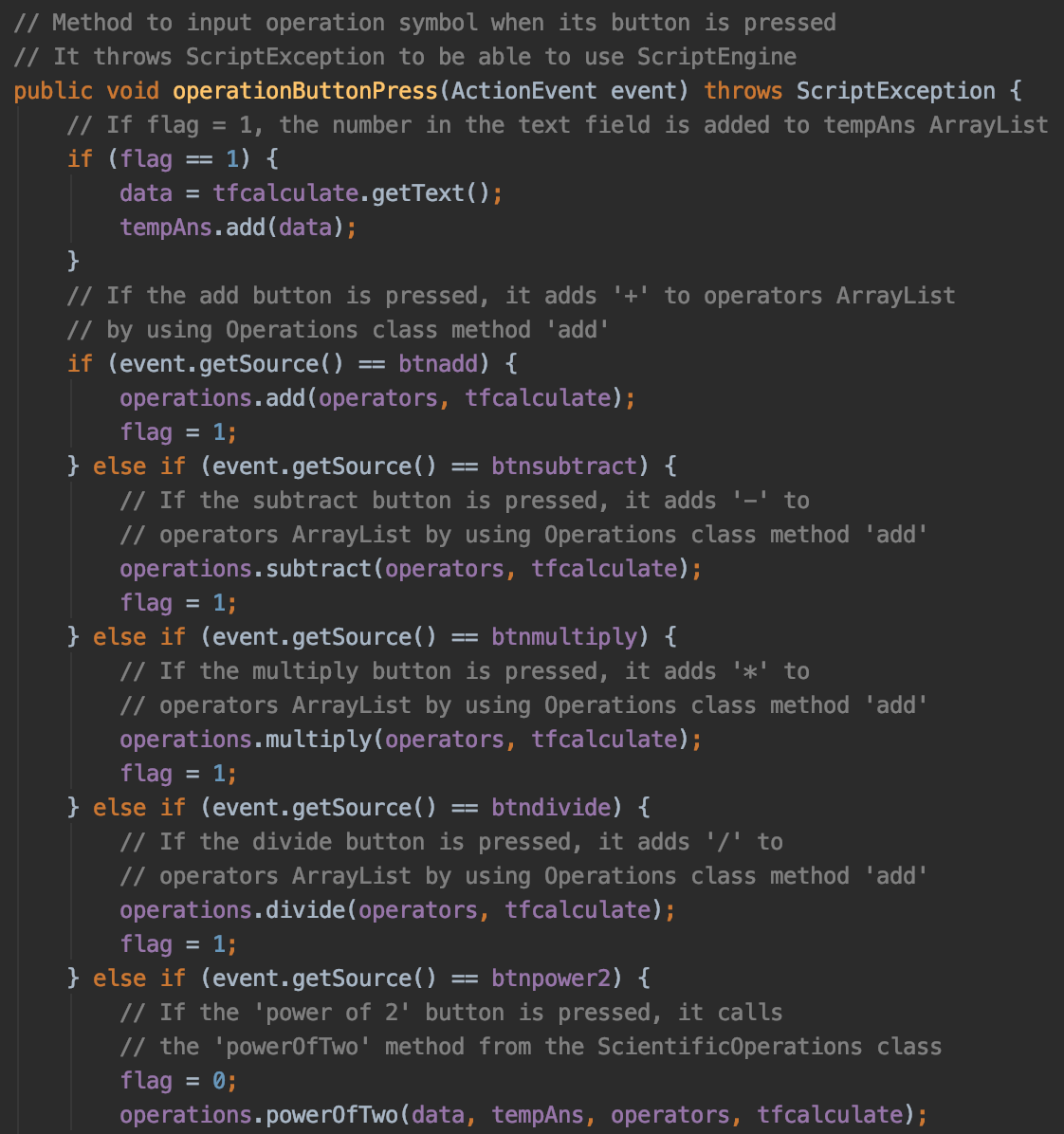
I import every packages I need for the code, like ActionEvent is used to receive events that happened with the buttons, Button, TextField and ArrayList are used to initialize the buttons and text field. ScriptEngine, ScriptEngineManager and ScriptException are used to evaluate a string for the solution of the math calculations.



Inside the class, I declared every attributes needed, like the buttons, text field, ‘data’ string to store the number input temporarily, ‘tempAns’ ArrayList to store the number inputs, ‘operators’ ArrayList to store mathematical operators and a flag to check certain conditions later on. I also declared an object named ‘operations’ from another class that I create called ‘ScientificOperations’ that I will explain later.



Inside the class I created a method called ‘buttonPress’ with event as the parameter. I assign this method to the number buttons, decimal point and clear button. In this method, if a number button is pressed, it will display the number on the ‘tfcalculate’ text field. If the decimal point button is pressed, it will display a decimal point on the text field. If the clear button is pressed, it will clear and delete the numbers on the text field.

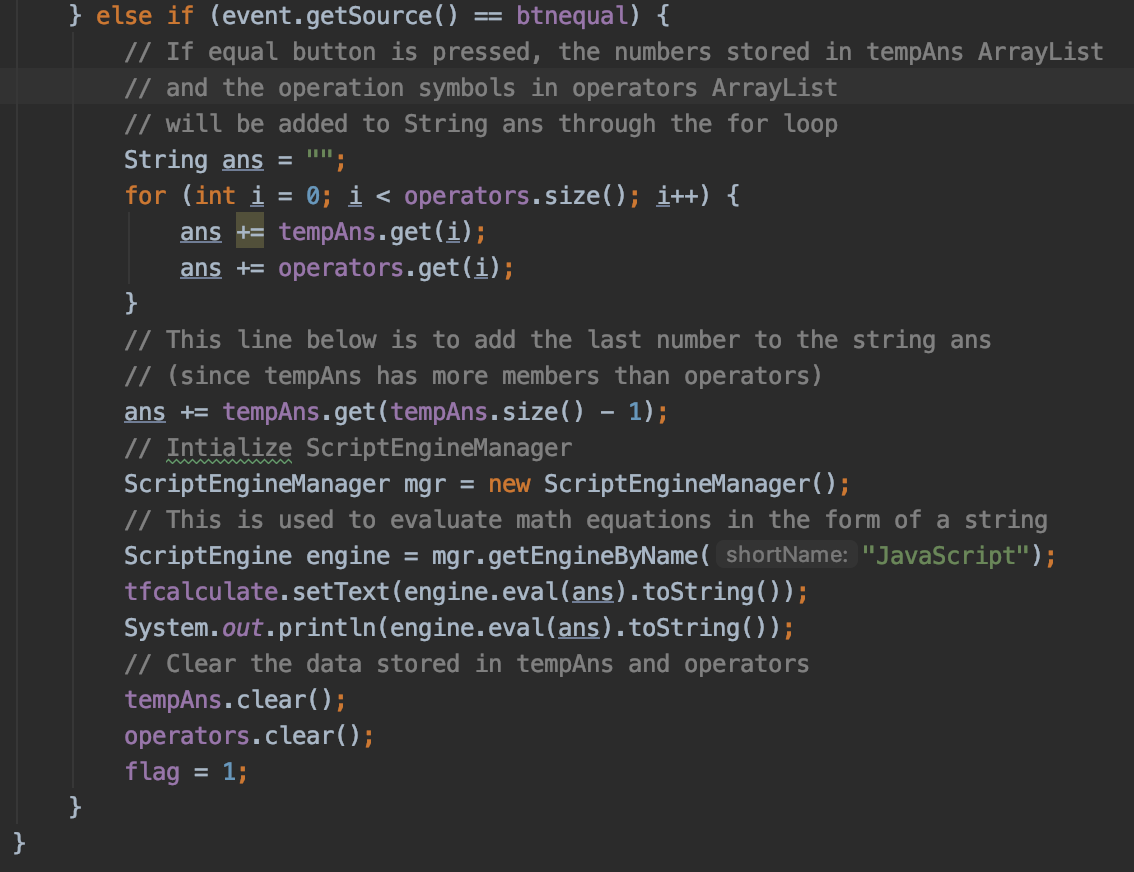


I also created a method called ‘operationButtonPress’ with event as the parameter. I assign this method to the mathematical operation buttons, like add, subtract, multiply, divide, square / power of two (x2) and equal buttons. The method uses ScriptException for the equal button. When the method is called, it will check if the flag is equal to 1, the numbers from the text field ‘tfcalculate’ will be stored as a string called ‘data’ and ‘data’ will be added to ‘tempAns’ ArrayList.

Then, if the add button is pressed, it will call the ‘add’ method from the ‘ScientificOperations’ class object ‘operations’ and sets the flag to 1. If the subtract button is pressed, it will call the ‘subtract’ method from the same object ‘operations’ and sets the flag to 1. The same thing applies to the multiply and divide button.

If the ‘power of two’ button is pressed, it will set the flag to 0 first to avoid error when we run the program. This is done because in the next line it calls the powerOfTwo method from ‘operations’ and when that method is called, it will add ‘\*’ (multiply) and the previous number.

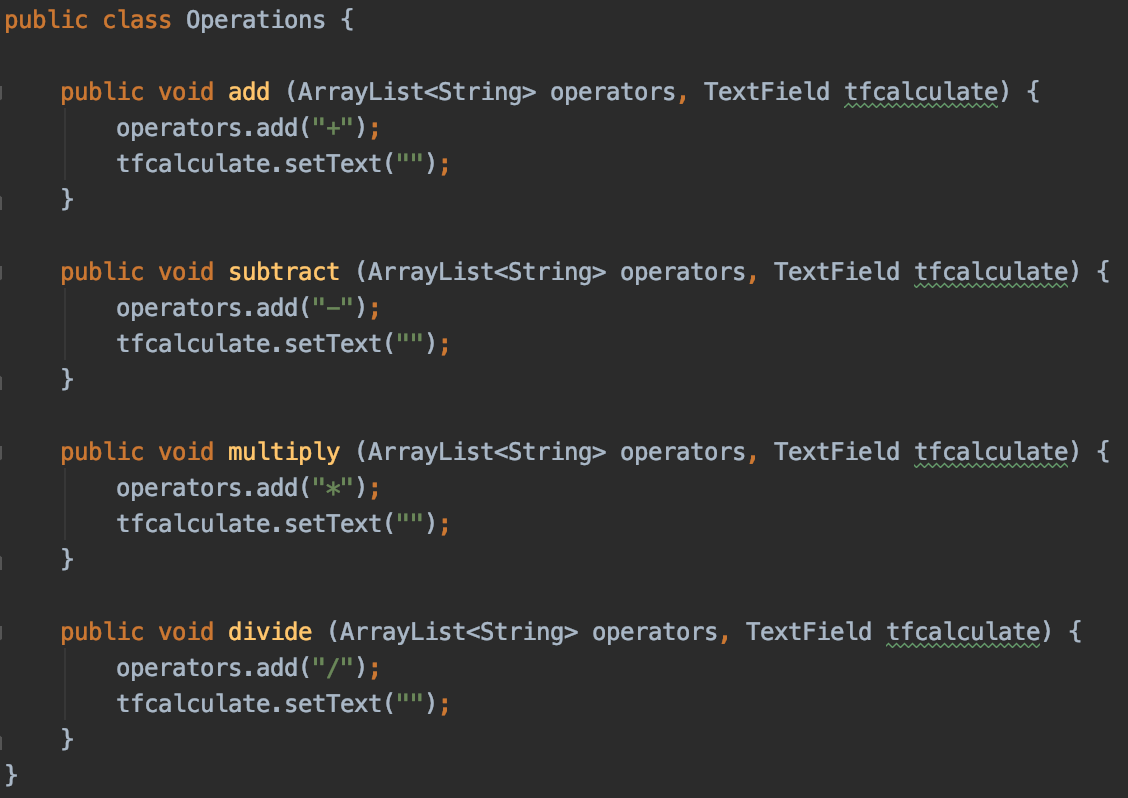
For example, the user wants to calculate ‘52 + 1’, so the user will input ‘5’ and press the power of two button, the program will automatically add ‘\*’ and ‘5’ to the calculation. Then, the user needs to press the add button but the text field is empty. Since the flag is equal to 0, it will not get the numbers on text field and it will not cause an error.



If the equal button is pressed, it will initialize a string called ‘ans’ to store the whole calculation. Then, numbers in ‘tempAns’ ArrayList and mathematical operation symbols in ‘operators’ ArrayList will be stored one by one into the string ‘ans’ consecutively. In the next line, the last element/number of ‘tempAns’ ArrayList is added to the string ‘ans’ because ‘tempAns’ will always have more members than ‘operators’. For example, the user wants to calculate ‘1 – 5 + 7’, ‘tempAns’ will have 3 elements (1, 5 and 7) and ‘operators’ will have 2 elements (- and +).

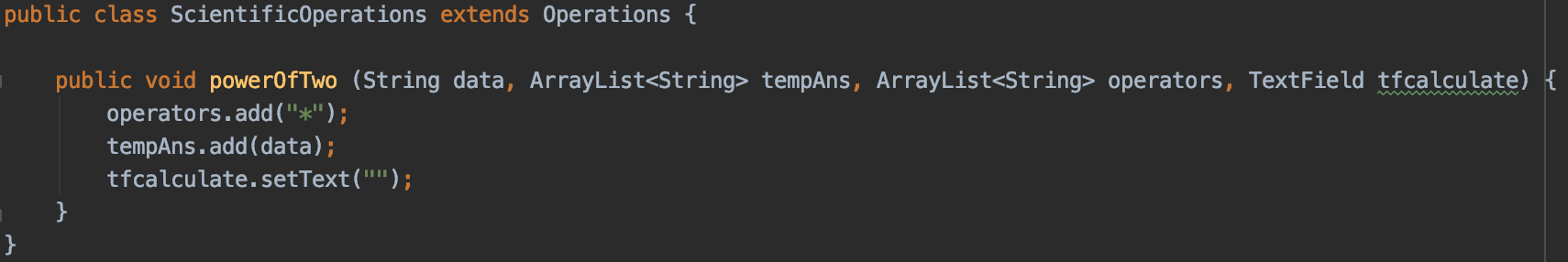
After the calculation is stored in ‘ans’, I initialize ScriptEngineManager and ScriptEngine. Then, the ScriptEngine will evaluate the string ‘ans’ and it will be shown on the ‘tfcalculate’ text field. Finally, clear ‘tempAns’ and ‘operators’ and set the flag to 1.

***Operations.java***



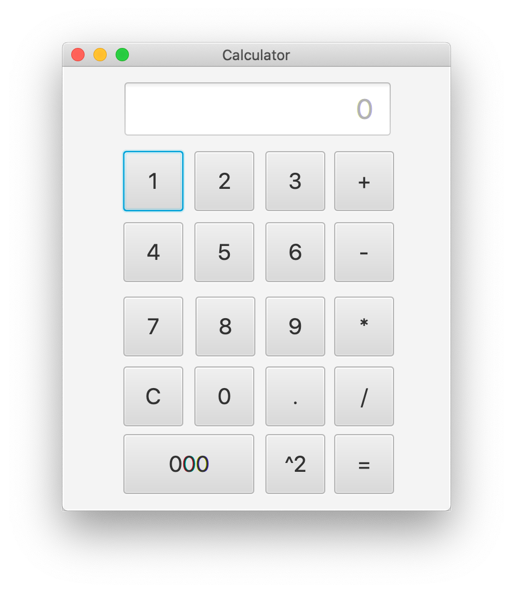
This class has methods that are used for the mathematical operation buttons. If the ‘add’ method is called, it adds ‘+’ to the ‘operators’ ArrayList and clears the ‘tfcalculate’ text field. If the ‘subtract’ method is called, it adds ‘-‘ to the ‘operators’ ArrayList and clears the ‘tfcalculate’ text field. The same thing applies to ‘multiply’ and ‘divide’ methods.

***ScientificOperations.java***

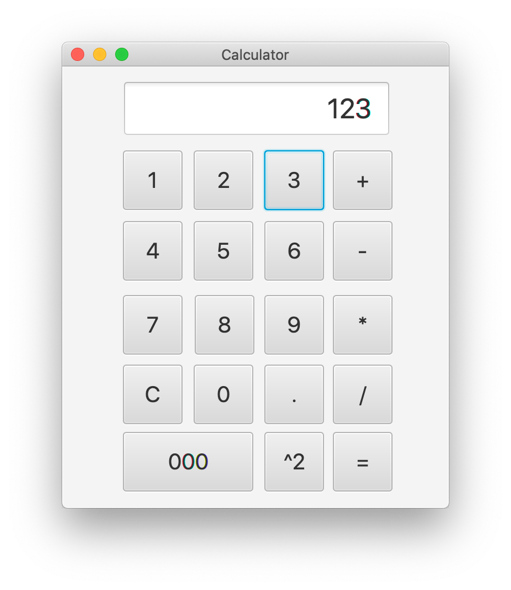


This class is the child class of ‘Operations’ class. I create a method inside this class called ‘powerOfTwo’. If it is called, it will add ‘\*’ to ‘operators’ ArrayList, it will also add the previous number inserted to ‘tempAns’ ArrayList, then clear ‘tfcalculate’ text field. The ‘operations’ class object is created from this class.

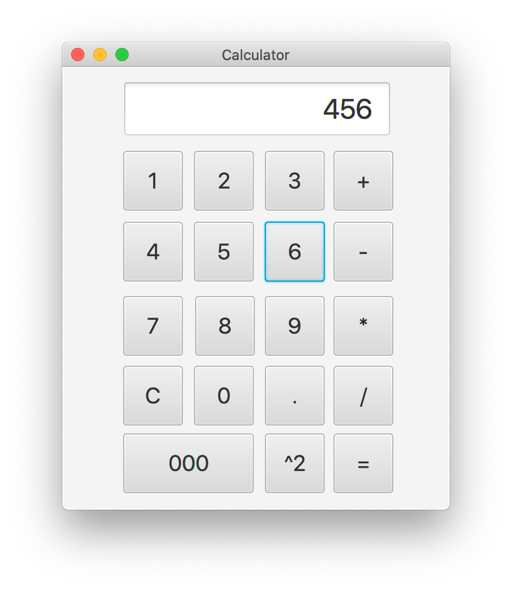
1. Evidence of Working Program



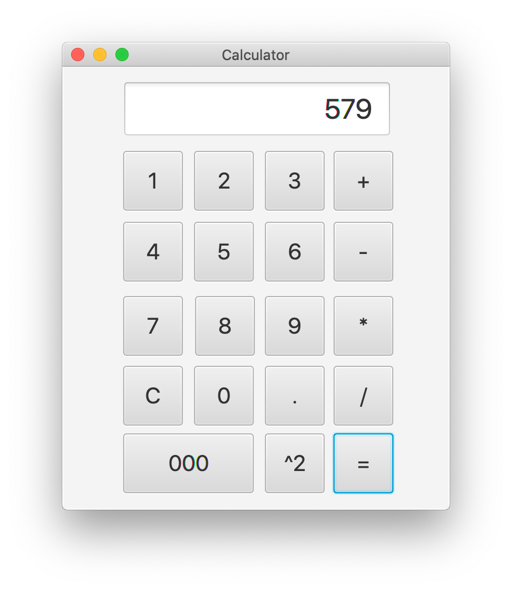
The appearance of the calculator GUI.



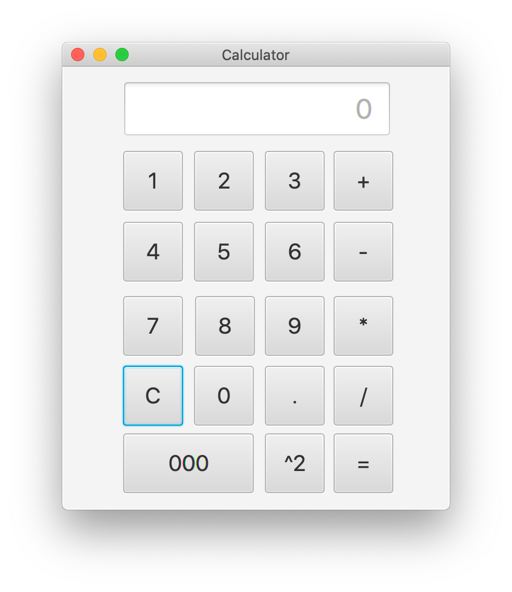
The appearance of the calculator GUI after the number ‘123’ is pressed.



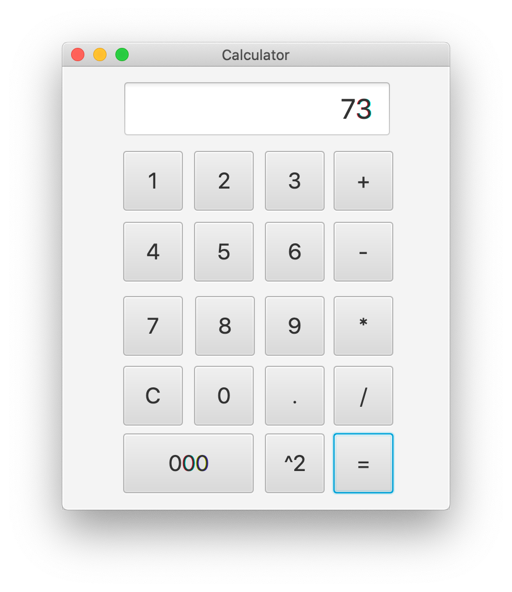
The appearance of the calculator GUI after ‘+’ button is pressed and the number ‘456’ is pressed.



The appearance of the calculator GUI after the ‘=’ button is pressed.



The appearance of the calculator GUI after the ‘C’ button is pressed.



The appearance of the calculator GUI after calculating ‘52+8\*6’.

1. Resources

* JavaFX basics: <https://www.jetbrains.com/help/idea/developing-a-javafx-application-examples.html>
* JavaFX calculator basics: <https://www.youtube.com/watch?v=r1qowt6yYm8&feature=emb_title>
* Solution of ‘how to evaluate a string’: <https://stackoverflow.com/questions/3422673/how-to-evaluate-a-math-expression-given-in-string-form>
* Fixing errors: https://stackoverflow.com