Southeast University

Project Report on

Simple Calculator Using JavaFx

Course Code: CSE 282

Course Title: Introduction to Programming Language II (JAVA)

Submitted to:

Ms. Lubna Yasmin Pinky
Guest Faculty, Department of CSE
Southeast University

Submitted by:

1. Md Mustakin Mahi
ID: 2020000000140
Department of CSE
Southeast University

2. Sharmin Sultana
ID: 2020000000123
Department of CSE
Southeast University

Table of Contents

Sl. no.	Content	Page no.
1	Project description	3
2	Project Overview	3-4
3	Features	5
4	Flow Chart	6
5	Coding Part	7-14
6	Output Part	15-19
7	Limitations	20
8	Future Scope of This Project	20

1. Project Description:

In this project we have made a Simple Calculator using core concept of Programming Language Java and a library of Java called “JavaFx”. We have designed a Graphical User Interface (GUI) in this project.

Using this calculator user can give input as his required digit and do required mathematical calculations like Addition, Subtraction, Multiplication and Division. We will add more functions in this project in future. Here we have used multiple class files to implement our project. We have used the Object-Oriented Programming (OOP) concept of Java that is taught in the course. The whole project, it’s functionalities, classes, source code and output are given below.

2. Project Overview:

Tools:

- Language: Java (Version: 17.0.1)
- Framework/Library: JavaFx
- IDE: NetBeans (Version 12.5)
- For GUI: Gluon Scene Builder

How we made it:

We opened a project in the IDE named “SimpCalc”. Then opened the primary.fxml file which automatically opened the Scene Builder for our GUI design. Then we designed the User Interface part to make it eye pleasant for the user. After finishing the designing part, we made the primary.fxml file as the controller and converted in to a

java class file named “PrimaryController.java”. We implemented the whole project in that file.

Even though we wrote all the codes in the PrimaryController.java file but the main execution of the program was done from the main file named “App.java”. We imported several packages under the main package named “com.mycompany.simpcalc”. Some of the imported packages are:
java.io.IOException,
javafx.fxml.FXMLLoader,
javafx.stage.Stage,
javafx.event.ActionEvent,
javafx.fxml.Initializable,
javafx.scene.control.Button;
javafx.scene.control.TextField etc.

We used **Method Overriding** that is a property of Java which allows a subclass or a child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes. After finishing all the user interface design and coding we implemented a simple calculator which will be shown at the later part of this document. We also used Switch-case function of basic programming concept in this project.

3. Features:

- One screen to display the digits that are given as input and the output.
- 16 buttons for the user to give different instructions like (0-9) digits, equal (=) button for result, clear (c) button for making the screen clear, four arithmetic signs to perform operations.
- **Basic functions:**
  - ✓ **Addition:** The addition operation can be done by clicking "+" button.
  - ✓ **Subtraction:** The subtraction operation can be done by clicking "-" button.
  - ✓ **Multiplication:** The multiplication operation can be done by clicking "*" button.
  - ✓ **Division:** The division operation can be done by clicking "/" button.
  - ✓ **Clearing display:** By clicking the "**c**" button, user can clear the screen.
  - ✓ **Result:** User can see the result on the display by clicking the "=" button.

Flow Chart

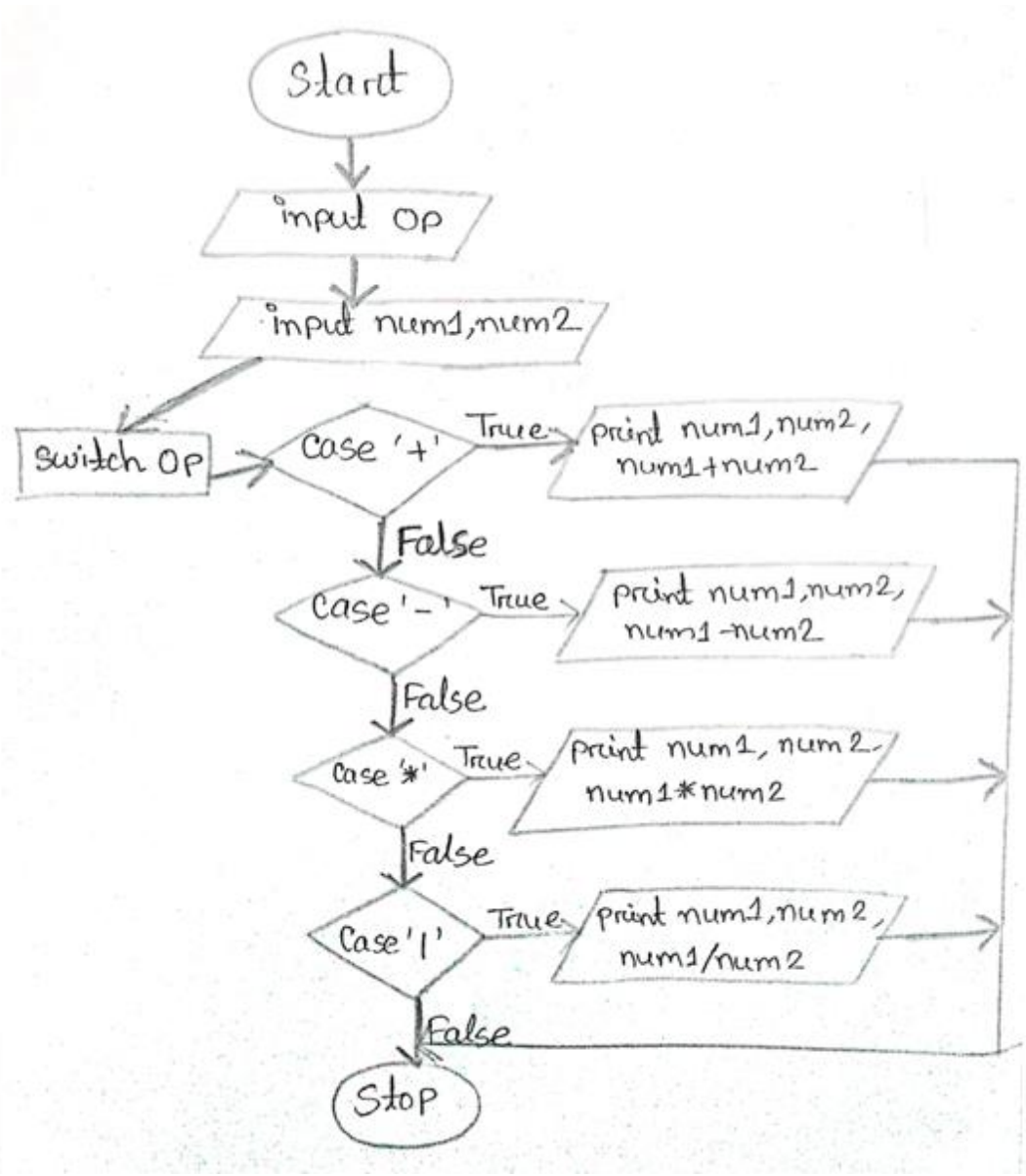

Fig: Flowchart of the program

Coding Part

App.java:

```
package com.mycompany.simpcalc;
import java.io.IOException;
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class App extends Application
{
        private static Scene scene;

    @Override
    public void start (Stage stage) throws IOException
    {
        scene = new Scene(loadFXML("primary"));
        stage.setScene(scene);
        stage.show();
    }

    static void setRoot(String fxml) throws IOException
    {
        scene.setRoot(loadFXML(fxml));
    }

    private static Parent loadFXML(String fxml) throws IOException
    {
        FXMLLoader fxmlLoader = new FXMLLoader(App.class.getResource(fxml + ".fxml"));
        return fxmlLoader.load();
    }

    public static void main (String [] args)
    {
        Launch ();
    }
}
```

PrimaryController.java:

```
package com.mycompany.simpcalc;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;

public class PrimaryController implements Initializable
{
    @FXML
    private Button btnPlus;
    @FXML
    private Button btnMinus;
    @FXML
    private Button btnMul;
    @FXML
    private Button btnDiv;
    @FXML
    private Button btnSeven;
    @FXML
    private Button btnSix;
    @FXML
    private Button btnNine;
    @FXML
    private Button btnEight;
    @FXML
    private Button btnThree;
    @FXML
    private Button btnTwo;
    @FXML
    private Button btnFive;
    @FXML
    private Button btnFour;
    @FXML
    private Button btnOne;
    @FXML
    private Button btnResult;
```

```
@FXML
private Button btnClear;
@FXML
private Button btnZero;
@FXML
private TextField display;

int num, num1, num2, Result;
String Op;
@Override
public void initialize (URL url, ResourceBundle rb)
{

}

@FXML
private void PlusClick(ActionEvent event)
{
   num1 = Integer.parseInt(display.getText());
   Op = "+";
   display.setText("0");
}

@FXML
private void MinusClick(ActionEvent event)
{
   num1 = Integer.parseInt(display.getText());
   Op = "-";
   display.setText("0");
}

@FXML
private void MulClick(ActionEvent event)
{
   num1 = Integer.parseInt(display.getText());
   Op = "*";
   display.setText("0");
}

@FXML
```

```
private void DivClick(ActionEvent event)
{
   num1 = Integer.parseInt(display.getText());
   Op = "/";
   display.setText("0");
}

@FXML
private void SevenClick(ActionEvent event)
{
   num = Integer.parseInt(display.getText());
   if (num == 0)
   {
      display.setText("7");
   }
   else
   {
      display.setText(display.getText()+ "7");
   }
}

@FXML
private void SixClick(ActionEvent event)
{
   num = Integer.parseInt(display.getText());
   if (num == 0)
   {
      display.setText("6");
   }
   else
   {
      display.setText(display.getText()+ "6");
   }
}

@FXML
private void NineClick(ActionEvent event)
{
   num = Integer.parseInt(display.getText());
   if (num == 0)
```

```
    {
        display.setText("9");
    }
    else
    {
        display.setText(display.getText()+ "9");
    }
}

@FXML
private void EightClick(ActionEvent event)
{
    num = Integer.parseInt(display.getText());
    if (num == 0)
    {
        display.setText("8");
    }
    else
    {
        display.setText(display.getText()+ "8");
    }
}

@FXML
private void ThreeClick(ActionEvent event)
{
    num = Integer.parseInt(display.getText());
    if (num == 0)
    {
        display.setText("3");
    }
    else
    {
        display.setText(display.getText()+ "3");
    }
}

@FXML
private void TwoClick(ActionEvent event)
{
```

```
    num = Integer.parseInt(display.getText());
    if (num == 0)
    {
        display.setText("2");
    }
    else
    {
        display.setText(display.getText()+ "2");
    }
}

@FXML
private void FiveClick(ActionEvent event)
{
    num = Integer.parseInt(display.getText());
    if (num == 0)
    {
        display.setText("5");
    }
    else
    {
        display.setText(display.getText()+ "5");
    }
}

@FXML
private void FourClick(ActionEvent event)
{
    num = Integer.parseInt(display.getText());
    if (num == 0)
    {
        display.setText("4");
    }
    else
    {
        display.setText(display.getText()+ "4");
    }
}

@FXML
```

```
private void OneClick (ActionEvent event)
{
   num = Integer.parseInt(display.getText());
   if (num == 0)
   {
      display.setText("1");
   }
   else
   {
      display.setText(display.getText()+ "1");
   }
}

@FXML
private void ResultClick(ActionEvent event)
{
   num2 = Integer.parseInt(display.getText());
   switch (Op)
   {
      case "+":
         Result = num1 + num2;
         break;
      case "-":
         Result = num1 - num2;
         break;
      case "*":
         Result = num1 * num2;
         break;
      case "/":
         Result = num1 / num2;
         break;

   }
   display.setText(String.valueOf(num1 + Op + num2 + "="+ Result));
}

@FXML
private void ClearClick (ActionEvent event)
{
   display.setText("0");
```

```
    }

    @FXML
    private void ZeroClick(ActionEvent event)
    {
        num = Integer.parseInt(display.getText());
       if (num == 0)
       {
          display.setText("0");
       }
       else
       {
          display.setText(display.getText()+ "0");
       }
    }

}
```

Output Part

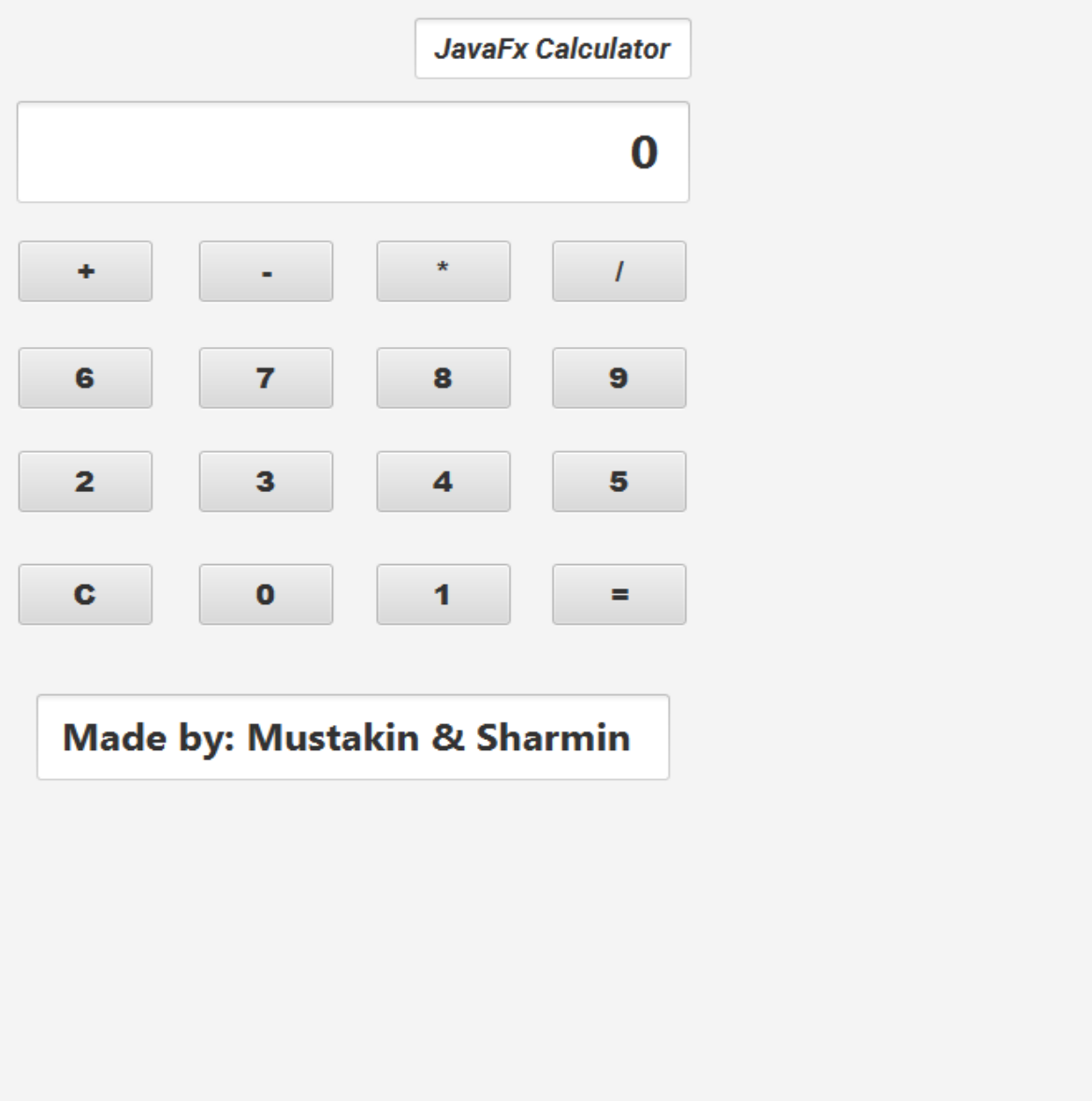

Fig: User Interface Part (No operation)

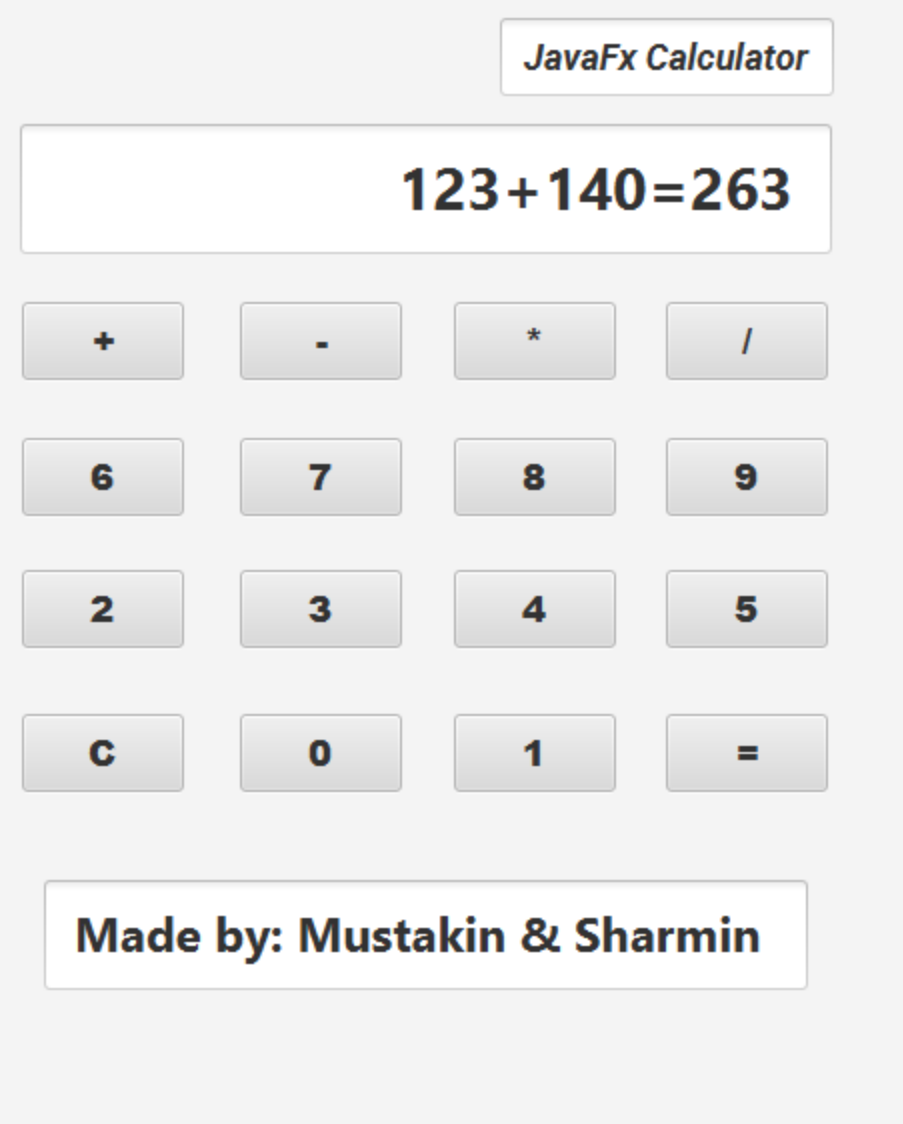

Fig: Addition Operation

Fig: Subtraction Operation

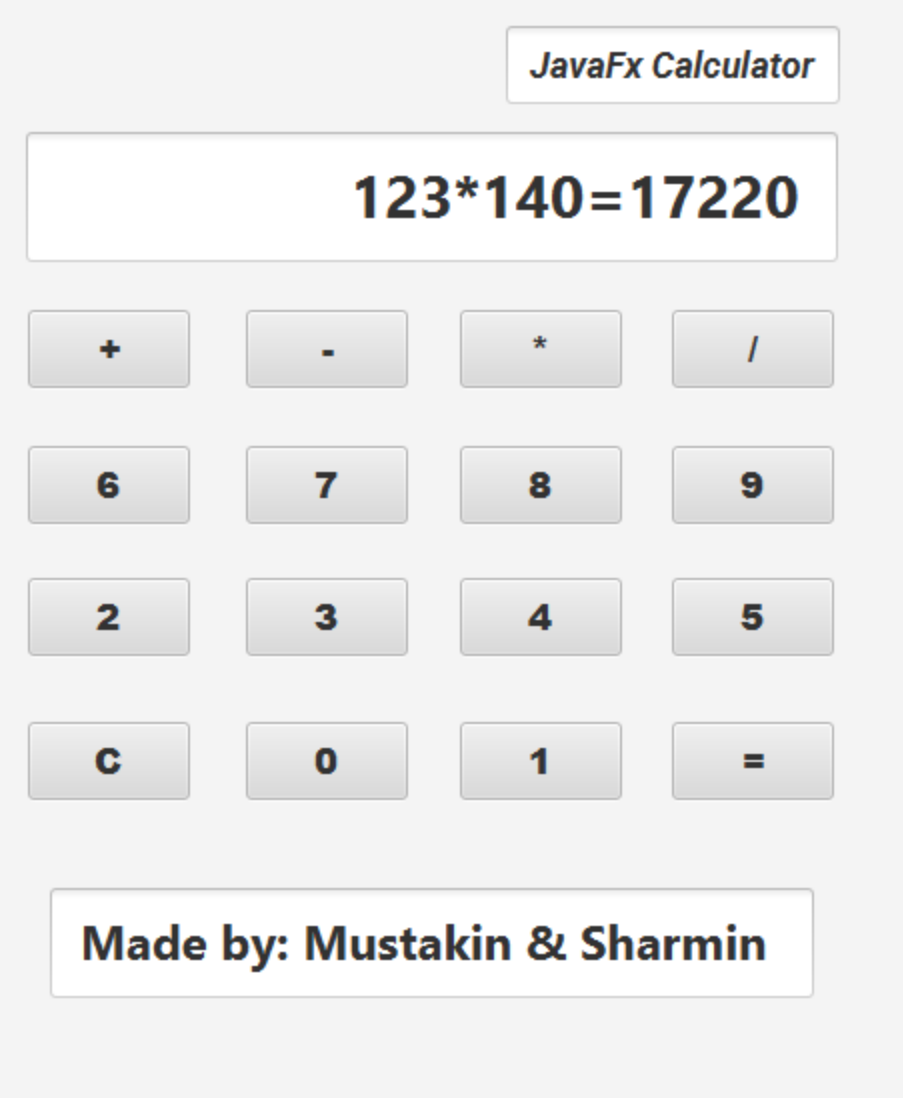

Fig: Multiplication Operation

Fig: Division Operation

Limitations:

- Our calculator only performs on integer numbers. It does not take floating number as input.
- It can not do higher level scientific operations.
- It only gives the integer part of the result as output in the division operation but does not give the floating points.

Future Scope of This Project:

- We have plan to add scientific features in this calculator.
- We will add CGPA calculation function in this calculator.
- We will try to over come the limitations of this project mentioned above.

THANK YOU