



KP14603 OBJECT ORIENTED PROGRAMMING
SEMESTER II
SESSION 19/20

ASSIGNMENT 2
INDIVIDUAL PROJECT

BASIC CALCULATOR

LECTURER : MADAM SITI HASNAH TANALOL

NAME	NO MATRIC
MUHAMMAD HAMSIAH BIN HAMDAN	BI19110153

Table of Contents

Introduction

Objectives

Java Codes

Object Oriented Concept Implementation

Read and Write Implementation

User Manual

Conclusion

Introduction

An calculator is typically a portable electronic device used to perform calculations, ranging from basic arithmetic to complex mathematics. The first solid state electronic calculator was created in early 1960s.

The project that I build is a simple calculator that only can do addition, subtraction, multiplication, and division. This calculator only can do two number operations only.

This project is to provide a better user friendly calculator that easy to use anytime, anywhere with a simple layout also make it easier for users. However, I build this friendly calculator because want to improve my understanding about the basic function of GUI and to understand more deeply object oriented programming.

Objectives

1. To make sure that all keys are correctly performing operation on the layout.
2. To design a simple calculator
3. To identify the requirements of the system.
4. To perform number of calculations in response to supplied input.

Java Codes

```
//Muhammad Hamsyah Bin Hamdan
```

```
//BI19110153
```

```
import java.awt.event.ActionEvent;
```

```
import javax.swing.JButton;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JLabel;
```

```
import javax.swing.JTextArea;
```

```
import javax.swing.SwingUtilities;
```

```
public class Calculator {
```

```
    private final JFrame frame = new JFrame("Calculator");
```

```
    private JTextArea txtScreen = new JTextArea();
```

```
    private final JLabel lblCredit = new JLabel("Created by Hamsyah Hamdan");
```

```
    //Calculation Variables:
```

```
    String strNum1 = "";
```

```
    String strNum2 = "";
```

```
    String[] statement;
```

```
    double num1 = 0;
```

```
    double num2 = 0;
```

```
    String strTotal = "";
```

```
    double totalVal = 0;
```

```
    private final JButton btnEqual = new JButton("=");
```

```
    //Number Buttons:
```

```
    private final JButton btn0 = new JButton("0");
```

```
    private final JButton btn1 = new JButton("1");
```

```
    private final JButton btn2 = new JButton("2");
```

```
    private final JButton btn3 = new JButton("3");
```

```
    private final JButton btn4 = new JButton("4");
```

```
    private final JButton btn5 = new JButton("5");
```

```
    private final JButton btn6 = new JButton("6");
```

```
    private final JButton btn7 = new JButton("7");
```

```
    private final JButton btn8 = new JButton("8");
```

```
    private final JButton btn9 = new JButton("9");
```

```
    //Operator Buttons:
```

```

private final JButton btnDiv = new JButton("/");
private final JButton btnMul = new JButton("*");
private final JButton btnSub = new JButton("-");
private final JButton btnAdd = new JButton("+");
private final JButton btnDec = new JButton(".");
private final JButton btnPower = new JButton("^");

public Calculator() {
    //Frame Attributes:
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);
    frame.setSize(400, 560);
    frame.setResizable(false);
    frame.setLayout(null);

    //txtScreen Attributes:
    txtScreen.setSize(380, 150);
    txtScreen.setLocation(7, 15);
    txtScreen.setEditable(false);

    //lblCredit Attributes:
    lblCredit.setSize(200, 100);
    lblCredit.setLocation(12, 150);

    //btn0:
    btn0.setSize(80, 50);
    btn0.setLocation(105, 470);
    btn0.addActionListener((ActionEvent arg0) -> {
        txtScreen.append("0");
    });

    //btn1:
    btn1.setSize(80, 50);
    btn1.setLocation(10, 395);
    btn1.addActionListener((ActionEvent arg0) -> {
        txtScreen.append("1");
    });

    //btn2:
    btn2.setSize(80, 50);
    btn2.setLocation(105, 395);
    btn2.addActionListener((ActionEvent arg0) -> {
        txtScreen.append("2");
    });

    //btn3:
    btn3.setSize(80, 50);

```

```
btn3.setLocation(200, 395);  
btn3.addActionListener((ActionEvent arg0) -> {  
    txtScreen.append("3");  
});
```

```
//btn4:  
btn4.setSize(80, 50);  
btn4.setLocation(10, 310);  
btn4.addActionListener((ActionEvent arg0) -> {  
    txtScreen.append("4");  
});
```

```
//btn5:  
btn5.setSize(80, 50);  
btn5.setLocation(105, 310);  
btn5.addActionListener((ActionEvent arg0) -> {  
    txtScreen.append("5");  
});
```

```
//btn6:  
btn6.setSize(80, 50);  
btn6.setLocation(200, 310);  
btn6.addActionListener((ActionEvent arg0) -> {  
    txtScreen.append("6");  
});
```

```
//btn7:  
btn7.setSize(80, 50);  
btn7.setLocation(10, 230);  
btn7.addActionListener((ActionEvent arg0) -> {  
    txtScreen.append("7");  
});
```

```
//btn8:  
btn8.setSize(80, 50);  
btn8.setLocation(105, 230);  
btn8.addActionListener((ActionEvent arg0) -> {  
    txtScreen.append("8");  
});
```

```
//btn9:  
btn9.setSize(80, 50);  
btn9.setLocation(200, 230);  
btn9.addActionListener((ActionEvent arg0) -> {  
    txtScreen.append("9");  
});
```

```

//btnDiv:
btnDiv.setSize(70, 60);
btnDiv.setLocation(310, 180);
btnDiv.addActionListener((ActionEvent arg0) -> {
    txtScreen.append("/");
});

//btnMul:
btnMul.setSize(70, 60);
btnMul.setLocation(310, 250);
btnMul.addActionListener((ActionEvent arg0) -> {
    txtScreen.append("*");
});

//btnSub
btnSub.setSize(70, 60);
btnSub.setLocation(310, 320);
btnSub.addActionListener((ActionEvent arg0) -> {
    txtScreen.append("-");
});

//btnAdd:
btnAdd.setSize(70, 60);
btnAdd.setLocation(310, 390);
btnAdd.addActionListener((ActionEvent arg0) -> {
    txtScreen.append("+");
});

//btnDec:
btnDec.setSize(80, 50);
btnDec.setLocation(200, 470);
btnDec.addActionListener((ActionEvent arg0) -> {
    txtScreen.append(".");
});

//btnPower:
btnPower.setSize(80, 50);
btnPower.setLocation(10, 470);
btnPower.addActionListener((ActionEvent arg0) -> {
    txtScreen.append("^");
});

//btnEqual:
btnEqual.setSize(70, 60);
btnEqual.setLocation(310, 460);
btnEqual.addActionListener((ActionEvent arg0) -> {
    if(txtScreen.getText().contains("+")){

```



```

        statement = txtScreen.getText().split("\\+");

        strNum1 = statement[0];
        strNum2 = statement[1];

        num1 = Double.parseDouble(strNum1);
        num2 = Double.parseDouble(strNum2);

        totalVal = num1 + num2;

        strTotal = Double.toString(totalVal);

        txtScreen.setText(strTotal);
    }
    else if(txtScreen.getText().contains("-")){
        statement = txtScreen.getText().split("\\-");

        strNum1 = statement[0];
        strNum2 = statement[1];

        num1 = Integer.parseInt(strNum1);
        num2 = Integer.parseInt(strNum2);

        totalVal = num1 - num2;

        strTotal = Double.toString(totalVal);

        txtScreen.setText(strTotal);
    }
    else if(txtScreen.getText().contains("/")){
        statement = txtScreen.getText().split("\\/");

        strNum1 = statement[0];
        strNum2 = statement[1];

        num1 = Double.parseDouble(strNum1);
        num2 = Double.parseDouble(strNum2);

        totalVal = num1 / num2;

        strTotal = Double.toString(totalVal);

        txtScreen.setText(strTotal);
    }
    else if(txtScreen.getText().contains("*")){
        statement = txtScreen.getText().split("\\*");

```

```

        strNum1 = statement[0];
        strNum2 = statement[1];

        num1 = Double.parseDouble(strNum1);
        num2 = Double.parseDouble(strNum2);

        totalVal = num1 * num2;

        strTotal = Double.toString(totalVal);

        txtScreen.setText(strTotal);
    }
    else if(txtScreen.getText().contains("^")){
        statement = txtScreen.getText().split("\\^");

        strNum1 = statement[0];
        strNum2 = statement[1];

        num1 = Double.parseDouble(strNum1);
        num2 = Double.parseDouble(strNum2);

        totalVal = 1;
        for(int i = 0; i < num2; i++){
            totalVal *= num1;
        }

        strTotal = Double.toString(totalVal);

        txtScreen.setText(strTotal);
    }
});

```

//Add Objects To Screen:

```

frame.add(txtScreen);
frame.add(btn0);
frame.add(btn1);
frame.add(btn2);
frame.add(btn3);
frame.add(btn4);
frame.add(btn5);
frame.add(btn6);
frame.add(btn7);
frame.add(btn8);
frame.add(btn9);

```

```

frame.add(lblCredit);

```

```
        frame.add(btnDiv);
        frame.add(btnMul);
        frame.add(btnSub);
        frame.add(btnAdd);
        frame.add(btnDec);
        frame.add(btnPower);

        frame.add(btnEqual);

        SwingUtilities.updateComponentTreeUI(frame);
    }

    public static void main(String[] args) {
        Calculator calculator;
        calculator = new Calculator();
    }
}
```

Object Oriented Concept Implementation

1. Object and Classes

- A class can be defined as a template/blueprint that describes the behavior/state that the object of its type support. It represents the set of properties or method that are common to all objects of one type.

```
public class Calculator {  
  
    private final JFrame frame = new JFrame("Calculator");  
    private JTextArea txtScreen = new JTextArea();  
    private final JLabel lblCredit = new JLabel("Created by Hamsyah Hamdan");  
  
    //Calculation Variables:  
    String strNum1 = "";  
    String strNum2 = "";  
    String[] statement;  
    double num1 = 0;  
    double num2 = 0;  
    String strTotal = "";  
    double totalVal = 0;
```

2. Interface

- The interface keyword is used to declare an interface. Here is a simple example to declare an interface.

- Implements ActionListener

```
//btn0:  
btn0.setSize(80, 50);  
btn0.setLocation(105, 470);  
btn0.addActionListener((ActionEvent arg0) -> {  
    txtScreen.append("0");  
});  
  
//btn1:  
btn1.setSize(80, 50);  
btn1.setLocation(10, 395);  
btn1.addActionListener((ActionEvent arg0) -> {  
    txtScreen.append("1");  
});  
  
//btn2:  
btn2.setSize(80, 50);  
btn2.setLocation(105, 395);  
btn2.addActionListener((ActionEvent arg0) -> {  
    txtScreen.append("2");
```

3. Inner Class

- In Java, it is also possible to nest classes (a class within a class). The purpose of nested classes is to group classes that belong together, which makes your code more readable and maintainable. To access the inner class, create an object of the outer class, and then create an object of the inner class.

```
public static void main(String[] args) {  
    Calculator calculator;  
    calculator = new Calculator();  
}
```

4. Abstraction

- A process of hiding the implementation details from the user, only the functionality will be provided to the user. In other words, the user will have the information on what the object does instead of how it does it.

```
private final JFrame frame = new JFrame("Calculator");  
private JTextArea txtScreen = new JTextArea();  
private final JLabel lblCredit = new JLabel("Created by Hamsyah Hamdan");  
  
//Calculation Variables:  
String strNum1 = "";  
String strNum2 = "";  
String[] statement;  
double num1 = 0;  
double num2 = 0;  
String strTotal = "";  
double totalVal = 0;
```

5. Encapsulation

- Encapsulation is one of the fundamental concepts in object oriented programming (OOP). It describes the idea of bundling data and methods that work on that within one unit.

```
public static void main(String[] args) {  
    Calculator calculator;  
    calculator = new Calculator();  
}
```

```
import java.awt.event.ActionEvent;  
  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JTextArea;  
import javax.swing.SwingUtilities;  
  
public class Calculator {
```

Read And Write Implementation

JTextArea is a part of java Swing package. It represents a multi line area that displays text. It is used to edit text. JTextArea inherits JComponent class. The text in JTextArea can be set to different available fonts and can be appended to new text. A text area can be customized to need of user.

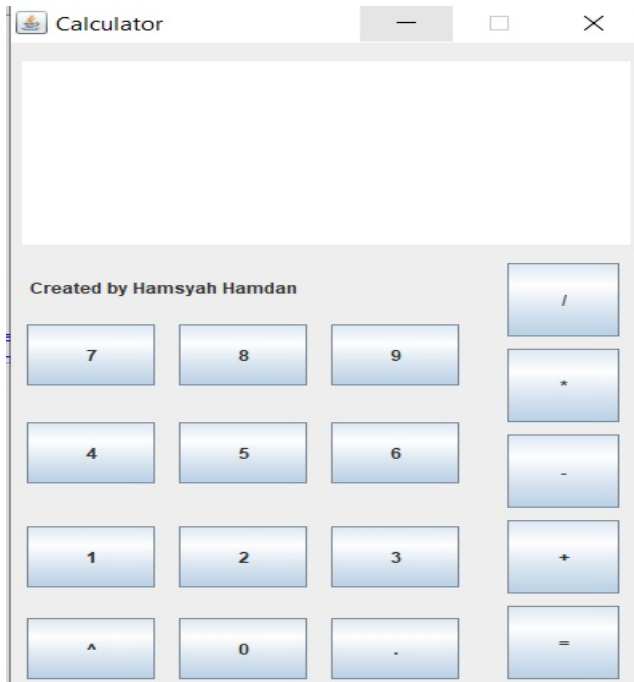
- JTextArea() : construct a new blank area.
- JTextArea(String s) : construct a new text area with given initial text.
- JTextArea(int row, int column) : Construct a new text area with a given number of rows and columns.
- JTextArea(String s, int row, int column) : construct a new text area with a given number of rows and columns and given initial text.

```
public class Calculator {  
  
    private final JFrame frame = new JFrame("Calculator");  
    private JTextArea txtScreen = new JTextArea();  
    private final JLabel lblCredit = new JLabel("Created by Hamsyah Hamdan");
```

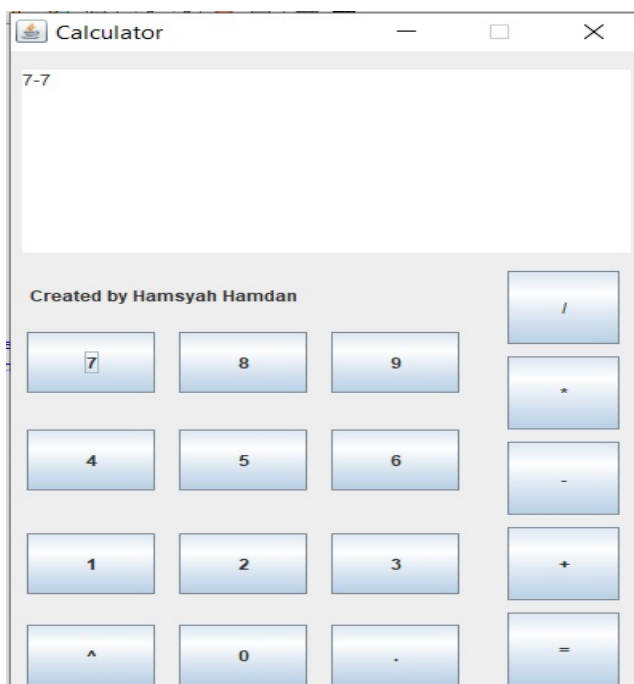
User Manual

This calculator is with a basic operation. For example operation subtraction, addition, divided and multiplication.

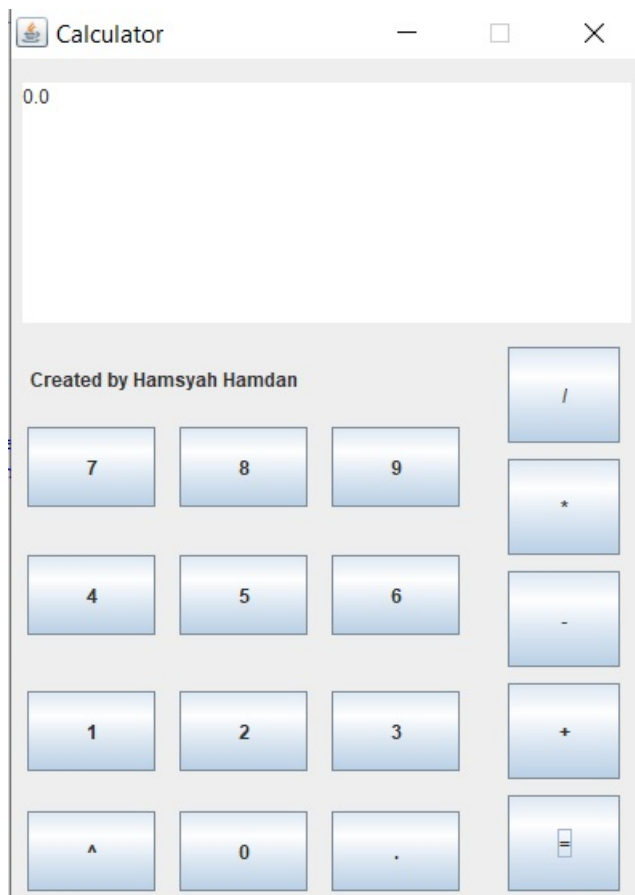
1. Run the project and you'll see this layout of GUI.



2. Add numbers with an operation
Example : 7-7



3. Then press symbol “=” (equal to), and you will get the answer of that basic question.



Conclusion

As a conclusion, the project that I had create is a simple calculator for a basic question mathematics, its easy to use. Also, its my first time make layout of GUI project. However, in this project is a good way to expand and get more understand deeply my knowledge about JAVA. Lastly, I'll keep improve this calculator system if there any more issues from other user.

