Assignment No. 07

Assignment Title: create a calculator using single function add(), sub() and multi().

Aim: Write a java program to create a calculator which performs addition, subtraction and multiplication of numbers for different types like integer, float and complex numbers using single function add(),sub() and multi().

Pre-Requisites: C/C++ Programming

Objective: We will learn how to build a simple calculator using Java AWT. This calculator has some simple functionality like all the basic mathematical operations and some special *addon features* using Java Swing components to create a simple calculator with only +, -, /, * operations.

Outcomes:

For creating a calculator, we can have the following different sets of input and output.

1. To Perform Addition:

When the addition expression "98+102" is typed,

it is expected that the result is displayed as "98+102=200.0".

2. To Perform Subtraction :

When the subtraction expression "200.0-58.75" is typed,

it is expected that the result is displayed as "200.0-58.75=141.25".

3. To Perform Multiplication :

When an multiplication expression "141.25*20" is typed,

it is expected that the result is displayed as "141.25*20=2825.0".

4. To Perform Division : When the denominator is non-zero

When an division expression "2825.0/5" is typed,

it is expected that the result is displayed as "2825.0/5=565.0".

5. To Perform Division : When the denominator is zero

When an division expression "565.0/0" is typed,

it is expected that the error is displayed as "565.0/0=Zero Divison Error".

Theory:

AWT(Abstract Window Toolkit) is an API that helps in building GUI (Graphical User Interface) based java applications. GUI helps in user interactions using some graphics. It primarily consists of a set of classes and methods that are required for creating and managing the GUI in a simplified manner such as buttons,windows,frame,textfield,RadioButton etc

In this article we will use Java AWT components to create a simple calculator with only +, -, /, * operations.

methods used :

1. add(Component c) : adds component to container.
2. addActionListenerListener(ActionListener d) : add actionListener for specified component
3. setBackground(Color c) : sets the background color of the specified container
4. setSize(int a, int b) : sets the size of container to specified dimensions.
5. setText(String s) : sets the text of the label to s.
6. getText() : returns the text of the label.

Algorithm/Steps:

LOGIC PART -I

1. FOR NUMERIC BUTTON. if(e. getSource()==b1){ //b1 for number 1 zt=l1. ...
2. FOR AIRTHMETIC BUTTON. if(e. getSource()==badd){ //FOR ADDITION num1=Double. ...
3. FOR EQUALS BUTTON. if(e. getSource()==bcalc){ num2=Double. ...
4. FOR CLEAR BUTTON. if(e. ...
5. FOR BACKSPACE BUTTON.

LOGIC PART -II

1. FOR NUMERIC BUTTON

if(e.getSource()==b1){ //b1 for number 1

```
zt=l1.getText();
 z=zt+"1";// 1 will merged at the end of the previous value
 l1.setText(z);
}
```

when any of the numeric button pressed
whatever value in label l1 will be stored in a variable zt and then concatenated with the corresponding number and and then displayed in the label l1
for NEGATIVE and DECIMAL PTS Button we did it similarly

2. FOR AIRTHMETIC BUTTON

```
if(e.getSource()==badd){  //FOR ADDITION
   num1=Double.parseDouble(l1.getText());
 z="";
 l1.setText(z);
 check=1; // 1 for the adddition
}
```

NOW here we store the value of label l1 into a variable num1 after converting into double type which will be technically 1st number
and then and set label l1 to null
we will just use a check variable for getting that this particular airthmetic button(here +) was clicked so we can do this operation in our = button

3. FOR EQUALS BUTTON

```
if(e.getSource()==bcalc){
   num2=Double.parseDouble(l1.getText());
 if(check==1)
   xd =num1+num2;
 if(check==2)
   xd =num1-num2;
 if(check==3)
   xd =num1*num2;
 if(check==4)
   xd =num1/num2;
 if(check==5)
   xd =num1%num2;
 l1.setText(String.valueOf(xd));
}
```

NOW again store the value of l1 into num2 variable which will be techincally 2nd number
and then check the value of variable check and then do corresponding operation and after that display result in label l1

4. FOR CLEAR BUTTON

```
if(e.getSource()==bclr){
  num1=0;
  num2=0;
  check=0;
  xd=0;
   z="";
  l1.setText(z);
   }
```

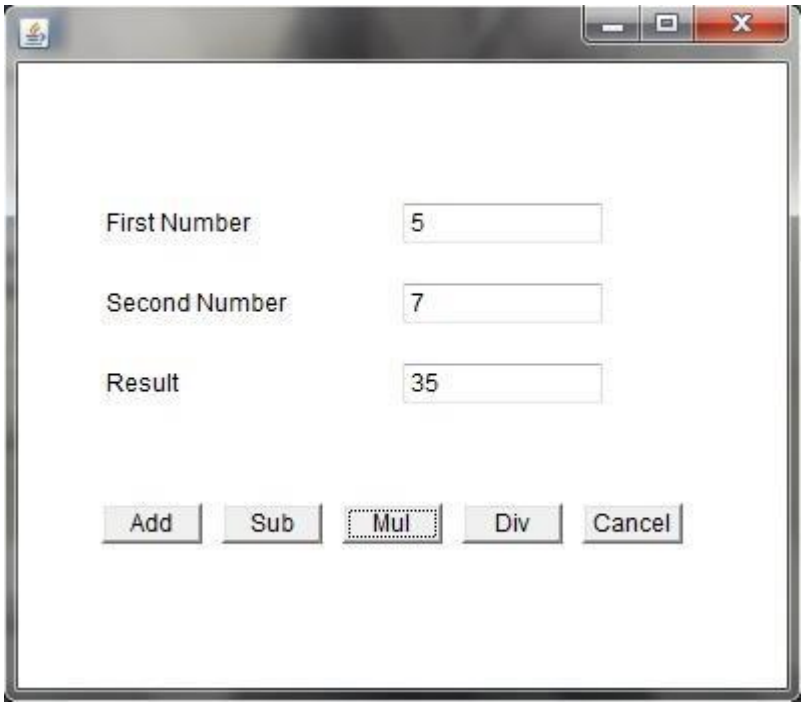
here updated all the variable we use to its Default value 0
and set label l1 to null so that we can start our new calculation afterward

5. FOR BACKSPACE BUTTON

```
if(e.getSource()==bback){  // FOR  BACKSPACE
  zt=l1.getText();
  try{
    z=zt.substring(0, zt.length()-1);
    }catch(StringIndexOutOfBoundsException f){return;}
  l1.setText(z);
}
```

here just updates the value in l1 by removing last digits using substring function
and handled one StringIndexOutOfBoundsException which occur when our value in label is null and still pressing back Button

sample Output:



Program Explanation:

1. When any button on the calculator is clicked, the function actionPerformed is called. Get the button clicked using getActionCommand.

2. If the button clicked is a digit or decimal point, then either of the following is executed :

a) The digit is concatenated to the the second number if no operator has been encountered.

b) Otherwise, the digit is concatenated to the first number.

3. If the button clicked is the clear button, then clear the input field.

4. If the button clicked is the equals operator, then either of the following is executed :

a) If neither the first number nor the second number is empty, then compute the result and display it in the frame

b) Otherwise, clear the input field.

5. If any of the button {+,-,*,/} is clicked, then either of the following is executed :

a) If either the operator or the second number is empty, then set the button clicked as the operator.

b) Otherwise, compute the result and display it in the frame.

Conclusion: Thus we have studied calculators in java with the help of AWT/Swing with event handling and perform addition, subtraction and multiplication .

Frequently Asked Questions:

1. What is the main purpose of AWT controls?
2. Which package in the AWT is used for event handling in java?
3. Which controls are supported by AWT in java?
4. How are the events handled in AWT?
5. What are the three main components of event handling?
6. What are the limitations of AWT?
7. Why AWT is abstract?
8. What is Swing used for?

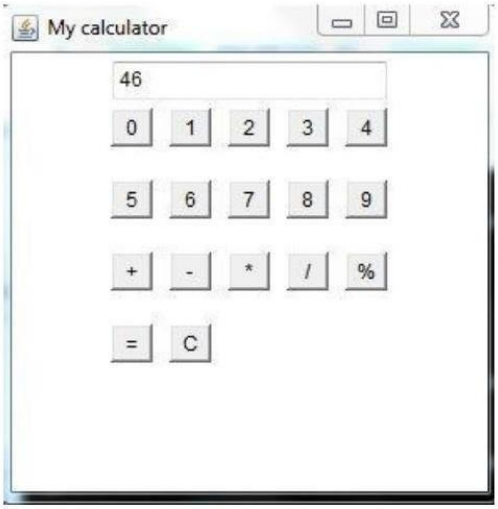
Code:

```
import java.awt.*;
import java.awt.event.*;
public class calculator implements ActionListener
{
    int c,n;
    String s1,s2,s3,s4,s5;
    Frame f;
    Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17;
    Panel p;
    TextField tf;
    GridLayout g;
    calculator()
    {
        f = new Frame("My calculator");
        p = new Panel();
        f.setLayout(new FlowLayout());
        b1 = new Button("0");
        b1.addActionListener(this);
        b2 = new Button("1");
        b2.addActionListener(this);
        b3 = new Button("2");
        b3.addActionListener(this);
        b4 = new Button("3");
        b4.addActionListener(this);
        b5 = new Button("4");
        b5.addActionListener(this);
        b6 = new Button("5");
        b6.addActionListener(this);
        b7 = new Button("6");
        b7.addActionListener(this);
        b8 = new Button("7");
        b8.addActionListener(this);
        b9 = new Button("8");
        b9.addActionListener(this);
        b10 = new Button("9");
        b10.addActionListener(this);
        b11 = new Button("+");
        b11.addActionListener(this);
        b12 = new Button("-");
        b12.addActionListener(this);
        b13 = new Button("*");
        b13.addActionListener(this);
        b14 = new Button("/");
        b14.addActionListener(this);
        b15 = new Button("%");
        b15.addActionListener(this);
        b16 = new Button("=");
        b16.addActionListener(this);
        b17 = new Button("C");
        b17.addActionListener(this);
        tf = new TextField(20);
```

```
        f.add(tf);
        g = new GridLayout(4,4,10,20);
        p.setLayout(g);
        p.add(b1);p.add(b2);p.add(b3);p.add(b4);p.add(b5);p.add(b6);p.add(b7);p.add(b8);p.
add(b9);
        p.add(b10);p.add(b11);p.add(b12);p.add(b13);p.add(b14);p.add(b15);p.add(b16);p.add
(b17);
        f.add(p);
        f.setSize(300,300);
        f.setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==b1)
        {
            s3 = tf.getText();
            s4 = "0";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b2)
        {
            s3 = tf.getText();
            s4 = "1";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b3)
        {
            s3 = tf.getText();
            s4 = "2";
            s5 = s3+s4;
            tf.setText(s5);
        }if(e.getSource()==b4)
        {
            s3 = tf.getText();
            s4 = "3";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b5)
        {
            s3 = tf.getText();
            s4 = "4";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b6)
        {
            s3 = tf.getText();
            s4 = "5";
            s5 = s3+s4;
            tf.setText(s5);
        }
```

```
        if(e.getSource()==b7)
        {
            s3 = tf.getText();
            s4 = "6";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b8)
        {
            s3 = tf.getText();
            s4 = "7";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b9)
        {
            s3 = tf.getText();
            s4 = "8";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b10)
        {
            s3 = tf.getText();
            s4 = "9";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b11)
        {
            s1 = tf.getText();
            tf.setText("");
            c=1;

        }
        if(e.getSource()==b12)
        {
            s1 = tf.getText();
            tf.setText("");
            c=2;

        }
        if(e.getSource()==b13)
        {
            s1 = tf.getText();
            tf.setText("");
            c=3;

        }
        if(e.getSource()==b14)
        {
            s1 = tf.getText();
            tf.setText("");
            c=4;
```

```
        }
        if(e.getSource()==b15)
        {
            s1 = tf.getText();
            tf.setText("");
            c=5;

        }
        if(e.getSource()==b16)
        {
            s2 = tf.getText();
            if(c==1)
            {
                n = Integer.parseInt(s1)+Integer.parseInt(s2);
                tf.setText(String.valueOf(n));
            }
            else
            if(c==2)
            {
                n = Integer.parseInt(s1)-Integer.parseInt(s2);
                tf.setText(String.valueOf(n));
            }
            else
            if(c==3)
            {
                n = Integer.parseInt(s1)*Integer.parseInt(s2);
                tf.setText(String.valueOf(n));
            }
            if(c==4)
            {
                try
                {
                    int p=Integer.parseInt(s2);
                    if(p!=0)
                    {
                                            n = Integer.parseInt(s1)/Integer.parseInt(s2);
                    tf.setText(String.valueOf(n));
                     }
                     else
                        tf.setText("infinite");

                }
                catch(Exception i){}
            }
            if(c==5)
            {
                n = Integer.parseInt(s1)%Integer.parseInt(s2);
                tf.setText(String.valueOf(n));
            }
        }
        if(e.getSource()==b17)
        {
            tf.setText("");
```




```
        }
    }

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