

| | |
|---------------|---------------------------------|
| EX.NO:10 | DESIGN OF SCIENTIFIC CALCULATOR |
| DATE:23.09.19 | |

AIM:

To Design a calculator using event-driven programming paradigm of Java with decimal manipulations and scientific manipulation

REQUIREMENTS:

Design a calculator using event-driven programming paradigm of Java with the following options.

- Decimal manipulations,
- Scientific manipulations.

ALGORITHM:

STEP1: Create a package calc.

STEP2: Create a class MyCalculator in the above package.

STEP3: Declare the buttons, panel, operand and operations as attributes.

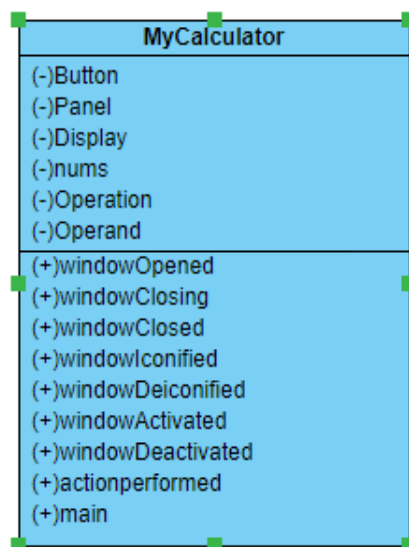
STEP4: Declare the constructors to pass the initial attributes.

STEP5: Import the required packages for the design of calculator.

STEP6: Provide the necessary methods for the decimal and scientific calculations.

STEP7: Stop.

CLASS DIAGRAM:



PROGRAM:

```

/**Developed by M. Uday kanth, EEE-B,212217105037
 *
 */
package cal;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
@SuppressWarnings("serial")
  
```

```

public class MyCalculator extends Frame implements WindowListener, ActionListener {
    TextField display;
    Button b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7,b_8,b_9;
    Button b_add,b_sub,b_mul,b_div,b_eq,b_sin,b_cos,b_tan,b_c;
    Panel p;
    String nums;
    Double op1,op2;
    int operator;
    static final int OP_ADD=1;
    static final int OP_SUB=2;
    static final int OP_MUL=3;
    static final int OP_DIV=4;
    public MyCalculator(){
        this.addWindowListener(this);
        this.setLayout(new GridLayout(2,2));
        nums="0";
        display=new TextField("0");
        display.setEditable(false);
        this.add(display);
        p=new Panel();
        p.setLayout(new GridLayout(3,2));
        this.add(p);
        b_0=new Button("0");
        b_0.addActionListener(this);
        p.add(b_0);
        b_1=new Button("1");
        b_1.addActionListener(this);
        p.add(b_1);
        b_2=new Button("2");
        b_2.addActionListener(this);
        p.add(b_2);
        b_3=new Button("3");
        b_3.addActionListener(this);
        p.add(b_3);
        b_4=new Button("4");
        b_4.addActionListener(this);
        p.add(b_4);
        b_5=new Button("5");
        b_5.addActionListener(this);
        p.add(b_5);
        b_6=new Button("6");
        b_6.addActionListener(this);
        p.add(b_6);
        b_7=new Button("7");
        b_7.addActionListener(this);
        p.add(b_7);
        b_8=new Button("8");

```

```

        b_8.addActionListener(this);
        p.add(b_8);
        b_9=new Button("9");
        b_9.addActionListener(this);
        p.add(b_9);
        b_add=new Button("+");
        b_add.addActionListener(this);
        p.add(b_add);
        b_sub=new Button("-");
        b_sub.addActionListener(this);
        p.add(b_sub);
        b_eq=new Button("=");
        b_eq.addActionListener(this);
        p.add(b_eq);
        b_div=new Button("/");
        b_div.addActionListener(this);
        p.add(b_div);
        b_mul=new Button("*");
        b_mul.addActionListener(this);
        p.add(b_mul);
        b_sin=new Button("sin");
        b_sin.addActionListener(this);
        p.add(b_sin);
        b_cos=new Button("cos");
        b_cos.addActionListener(this);
        p.add(b_cos);
        b_tan=new Button("tan");
        b_tan.addActionListener(this);
        p.add(b_tan);
        b_c=new Button("clear");
        b_c.addActionListener(this);
        p.add(b_c);
    }
    public static void main(String[] args) {
        MyCalculator mc;
        mc=new MyCalculator();
        mc.setTitle("Calculator");
        mc.setSize(400,300);
        mc.setVisible(true);
    }
    @Override
    public void windowOpened(WindowEvent e) {}
    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
    @Override

```

```

public void windowClosed(WindowEvent e) {}
@Override
public void windowIconified(WindowEvent e) {}
@Override
public void windowDeiconified(WindowEvent e) {}
@Override
public void windowActivated(WindowEvent e) {}
@Override
public void windowDeactivated(WindowEvent e) {}
@Override
public void actionPerformed(ActionEvent e) {
    if(e.getSource()==b_0)
    {
        nums=nums+"0";
        display.setText(nums);
    }else if(e.getSource()==b_1){
        nums=nums+"1";
        display.setText(nums);
    }else if(e.getSource()==b_2){
        nums=nums+"2";
        display.setText(nums);
    }else if(e.getSource()==b_3){
        nums=nums+"3";
        display.setText(nums);
    }else if(e.getSource()==b_4){
        nums=nums+"4";
        display.setText(nums);
    }else if(e.getSource()==b_5){
        nums=nums+"5";
        display.setText(nums);
    }else if(e.getSource()==b_6){
        nums=nums+"6";
        display.setText(nums);
    }else if(e.getSource()==b_7){
        nums=nums+"7";
        display.setText(nums);
    }else if(e.getSource()==b_8){
        nums=nums+"8";
        display.setText(nums);
    }else if(e.getSource()==b_9){
        nums=nums+"9";
        display.setText(nums);
    }else if(e.getSource()==b_add){
        op1=Double.parseDouble(nums);
        nums="0";
        display.setText(nums);
        operator=OP_ADD;
    }
}

```

```

}else if(e.getSource()==b_sub){
    op1=Double.parseDouble(nums);
    nums="0";
    display.setText(nums);
    operator=OP_SUB;

}else if(e.getSource()==b_mul){
    op1=Double.parseDouble(nums);
    nums="0";
    display.setText(nums);
    operator=OP_MUL;
}else if(e.getSource()==b_div){
    op1=Double.parseDouble(nums);
    nums="0";
    display.setText(nums);
    operator=OP_DIV;
}else if(e.getSource()==b_eq){
    op2=Double.parseDouble(nums);
    switch(operator){
        case 1:
            nums=""+(op1+op2);
            display.setText(nums);
            break;
        case 2:
            nums=""+(op1-op2);
            display.setText(nums);
            break;
        case 3:
            nums=""+(op1*op2);
            display.setText(nums);
            break;
        case 4:
            nums=""+(op1/op2);
            display.setText(nums);
            break;
    }
    display.setText(nums);
}else if(e.getSource()==b_sin){
    op1=Double.parseDouble(nums);
    nums="" + Math.sin(op1*Math.PI/180);
    display.setText(nums);
}else if(e.getSource()==b_cos){
    op1=Double.parseDouble(nums);
    nums="" + Math.cos(op1*Math.PI/180);
    display.setText(nums);
}
else if(e.getSource()==b_tan){

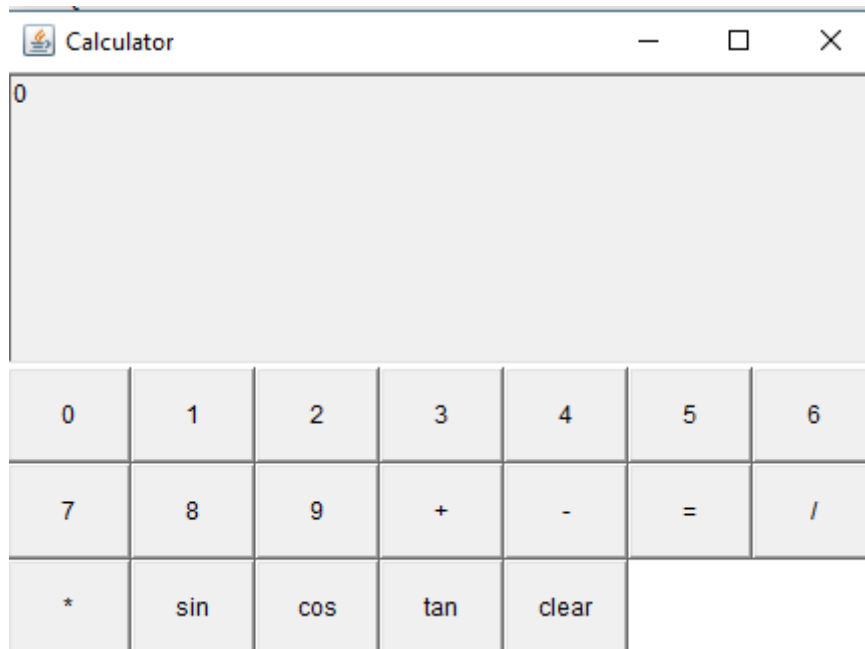
```

```

        op1=Double.parseDouble(nums);
        nums="" + Math.cos(op1*Math.PI/180);
        display.setText(nums);
    }
    else if(e.getSource()==b_c){
        nums=0+"";
        display.setText(nums);
    }
}
}
}

```

OUTPUT:



RESULT:

To Design a calculator using event-driven programming paradigm of Java with decimal manipulations and scientific manipulation

