



UMS
UNIVERSITI MALAYSIA SABAH

FACULTY OF COMPUTING AND INFORMATICS

HC 00 SOFTWARE ENGINEERING

SEMESTER 2 (2019 / 2020)

KK 14203 OBJECT ORIENTED PROGRAMMING

SECTION 1

INDIVIDUAL ASSIGNMENT

PROJECT 2 (CALCULATOR)

PREPARED BY

AMANINA IZATI BINTI AMRAN (BI19110148)

PREPARED FOR

DR. MOHD SHAMRIE SAININ

1.0 INTRODUCTION

The calculator is an extension of a mathematician and it has opened up new possibilities within mathematics. It is a machine though and it is only capable of doing what it is programmed to do. Accordingly, this project aims to develop the internal programmed computational code in the form of a computer program that a scientific calculator could use to compute functions such as square root. The idea of this project assumes that the programmer has already developed the very basic addition, subtraction, multiplication, division and integer splicing functions. Then using these basic functions, the program will then compute other more complicated functions found on typical scientific calculator.

2.0 JAVA CODE

```
import java.awt.*;
import java.awt.event.*;

public class MyCalculator extends Frame
{

    public boolean setClear=true;
    double number, memValue;
    char op;

    String digitButtonText[] = {"7", "8", "9", "4", "5", "6",
    "1", "2", "3", "0", "+/-", "." };
    String operatorButtonText[] = {"/", "sqrt", "*", "%", "-",
    "+", "=" };
    String specialButtonText[] = {"Backspc", "C", "CE" };

    MyDigitButton digitButton[]=new
MyDigitButton[digitButtonText.length];
    MyOperatorButton operatorButton[]=new
```

```

MyOperatorButton[operatorButtonText.length];
    MySpecialButton specialButton[]=new
MySpecialButton[specialButtonText.length];

    Label displayLabel=new Label("0",Label.RIGHT);
    Label memLabel=new Label(" ",Label.RIGHT);

    final int FRAME_WIDTH=325,FRAME_HEIGHT=325;
    final int HEIGHT=30, WIDTH=30, H_SPACE=10,V_SPACE=10;
    final int TOPX=30, TOPY=50;

    MyCalculator(String frameText)
    {
        super(frameText);

        int tempX=TOPX, y=TOPY;
        displayLabel.setBounds(tempX,y,240,HEIGHT);
        displayLabel.setBackground(Color.BLACK);
        displayLabel.setForeground(Color.WHITE);
        add(displayLabel);

        memLabel.setBounds(TOPX, TOPY+HEIGHT+ V_SPACE,WIDTH,
HEIGHT);
        add(memLabel);
        tempX=TOPX+1*(WIDTH+H_SPACE); y=TOPY+1*(HEIGHT+V_SPACE);

        for(int i=0;i<specialButton.length;i++)
        {
            specialButton[i]=new
MySpecialButton(tempX,y,WIDTH*2,HEIGHT,specialButtonText[i],
this);

            specialButton[i].setForeground(Color.RED);

```

```

        tempX=tempX+2*WIDTH+H_SPACE;
    }

    int digitX=TOPX+WIDTH+H_SPACE;
    int digitY=TOPY+2*(HEIGHT+V_SPACE);
    tempX=digitX;  y=digitY;

    for(int i=0;i<digitButton.length;i++)
    {
        digitButton[i]=new
MyDigitButton(tempX,y,WIDTH,HEIGHT,digitButtonText[i], this);
        digitButton[i].setForeground(Color.BLUE);
        tempX+=WIDTH+H_SPACE;

        if((i+1)%3==0){tempX=digitX; y+=HEIGHT+V_SPACE;}
    }

    int opsX=digitX+2*(WIDTH+H_SPACE)+H_SPACE;
    int opsY=digitY;
    tempX=opsX;  y=opsY;

    for(int i=0;i<operatorButton.length;i++)
    {
        tempX+=WIDTH+H_SPACE;
        operatorButton[i]=new
MyOperatorButton(tempX,y,WIDTH,HEIGHT,operatorButtonText[i],
this);
        operatorButton[i].setForeground(Color.RED);

        if((i+1)%2==0){tempX=opsX; y+=HEIGHT+V_SPACE;}
    }

```

```

addWindowListener(
    new WindowAdapter()
    {
        public void windowClosing(WindowEvent ev)
        {System.exit(0);}
    });

setLayout(null);
setSize(FRAME_WIDTH,FRAME_HEIGHT);
setVisible(true);
}

static String getFormattedText(double temp)
{
    String resText=""+temp;
    if(resText.lastIndexOf(".0")>0)
        resText=resText.substring(0,resText.length()-2);
    return resText;
}

public static void main(String []args)
{
    new MyCalculator("Calculator");
}
}

class MyDigitButton extends Button implements ActionListener
{
    MyCalculator cl;

    MyDigitButton(int x,int y, int width,int height,String cap,
MyCalculator clc)

```

```

{
    super(cap);
    setBounds(x,y,width,height);
    this.cl=clc;
    this.cl.add(this);
    addActionListener(this);
}

static boolean isInString(String s, char ch)
{
    for(int i=0; i<s.length();i++)
        if(s.charAt(i)==ch)
            return true;
    return false;
}

public void actionPerformed(ActionEvent ev)
{
    String
tempText=((MyDigitButton)ev.getSource()).getLabel();

    if(tempText.equals("."))
    {
        if(cl.setClear)
        {cl.displayLabel.setText("0.");cl.setClear=false;}
        else if(!isInString(cl.displayLabel.getText(), '.'))

cl.displayLabel.setText(cl.displayLabel.getText()+".");
        return;
    }

    int index=0;

```

```

        try{
            index=Integer.parseInt(tempText);
        }catch (NumberFormatException e){
            return;}

        if (index==0 && cl.displayLabel.getText().equals("0"))
            return;

        if(cl.setClear)
            {cl.displayLabel.setText(""+index);cl.setClear=false;}
        else

cl.displayLabel.setText(cl.displayLabel.getText()+index);
    }
}

class MyOperatorButton extends Button implements ActionListener
{
    MyCalculator cl;

    MyOperatorButton(int x,int y, int width,int height,String
cap, MyCalculator clc)
    {
        super(cap);
        setBounds(x,y,width,height);
        this.cl=clc;
        this.cl.add(this);
        addActionListener(this);
    }

    public void actionPerformed(ActionEvent ev)

```

```

{
    String
    opText= ( (MyOperatorButton)ev.getSource() ).getLabel();

    cl.setClear=true;
    double temp=Double.parseDouble(cl.displayLabel.getText());

    if(opText.equals("sqrt"))
    {
        try
        {double tempd=Math.sqrt(temp);

cl.displayLabel.setText(MyCalculator.getFormattedText(tempd));}
        catch(ArithmeticException excp)
        {cl.displayLabel.setText("Divide by 0.");}
        return;
    }
    if(!opText.equals("="))
    {
        cl.number=temp;
        cl.op=opText.charAt(0);
        return;
    }
    switch(cl.op)
    {
        case '+':
            temp+=cl.number;
            break;
        case '-':
            temp=cl.number-temp;
            break;
        case '*':

```



```

        temp*=cl.number;
        break;
    case '%':
        try{temp=cl.number%temp;}
        catch(ArithmeticException excp)
        {cl.displayLabel.setText("Divide by 0.");
         return;}
        break;
    case '/':
        try{temp=cl.number/temp;}
        catch(ArithmeticException excp)
        {cl.displayLabel.setText("Divide by 0.");
         return;}
        break;
    }

    cl.displayLabel.setText(MyCalculator.getFormattedText(temp));
}

}

class MyMemoryButton extends Button implements ActionListener
{
    MyCalculator cl;

    MyMemoryButton(int x,int y, int width,int height,String cap,
MyCalculator clc)
    {
        super(cap);
        setBounds(x,y,width,height);
        this.cl=clc;
        this.cl.add(this);
        addActionListener(this);
    }
}

```

```

    }

    public void actionPerformed(ActionEvent ev)
    {
        char
memop=( (MyMemoryButton) ev.getSource() ).getLabel().charAt(1);
        cl.setClear=true;
        double temp=Double.parseDouble(cl.displayLabel.getText());

        switch(memop)
        {
            case 'C':
                cl.memLabel.setText(" ");cl.memValue=0.0;
                break;
            case 'R':

cl.displayLabel.setText(MyCalculator.getFormattedText(cl.memValue));

                break;
            case 'S':
                cl.memValue=0.0;
            case '+':

cl.memValue+=Double.parseDouble(cl.displayLabel.getText());
                if(cl.displayLabel.getText().equals("0") ||
cl.displayLabel.getText().equals("0.0") )
                    cl.memLabel.setText(" ");
                else
                    cl.memLabel.setText("M");
                break;
        }
    }
}

```

```
}
```

```
class MySpecialButton extends Button implements ActionListener
{
    MyCalculator cl;

    MySpecialButton(int x,int y, int width,int height,String cap,
MyCalculator clc)
    {
        super(cap);
        setBounds(x,y,width,height);
        this.cl=clc;
        this.cl.add(this);
        addActionListener(this);
    }

    static String backSpace(String s)
    {
        String Res="";
        for(int i=0; i<s.length()-1; i++) Res+=s.charAt(i);
        return Res;
    }

    public void actionPerformed(ActionEvent ev)
    {
        String
opText=((MySpecialButton)ev.getSource()).getLabel();

        if(opText.equals("Backspc"))
        {
            String tempText=backSpace(cl.displayLabel.getText());
            if(tempText.equals(""))
```

```

        cl.displayLabel.setText("0");
    else
        cl.displayLabel.setText(tempText);
    return;
}
if(opText.equals("C"))
{
    cl.number=0.0; cl.op=' '; cl.memValue=0.0;
    cl.memLabel.setText(" ");
}

cl.displayLabel.setText("0");cl.setClear=true;
}
}

```

3.0 OBJECT ORIENTED CONCEPT IMPLEMENTATION

The calculator has many actions. Several keys (Backspace, +, /, -, *, C) cause the input action to be done and then go the state waiting for the calculation acknowledgment. The state machine returns to the state either directly if the calculation result is OK or error state if the calculation is erroneous, there are various different errors possible. The state machine waits in the error states for keys : C and / or CE returning eventually to the state.

The extension to operand functions such as sqrt is relatively easy. These functions are simple calculations. It is call corresponding functions as input actions in the state. The operand functions are treated as other functions, they trigger an input action in the present state and go to the busy state where they wait for an calculation result.

4.0 READ AND WRITE IMPLEMENTATION

i. Read Implementation

```
        catch(ArithmeticException excp)
        {cl.displayLabel.setText("Divide by 0.");}
        return;
    }
    if(opText.equals("sqrt"))
    {
        try
        {double tempd=Math.sqrt(temp);

cl.displayLabel.setText(MyCalculator.getFormattedText(tempd
));}

        catch(ArithmeticException excp)
        {cl.displayLabel.setText("Divide by 0.");}
        return;
    }
    if(!opText.equals("="))
    {
        cl.number=temp;
        cl.op=opText.charAt(0);
        return;
    }
    switch(cl.op)
    {
        case '+':
            temp+=cl.number;
            break;
        case '-':
            temp=cl.number-temp;
            break;
```

```

        case '*':
            temp*=cl.number;
            break;
        case '%':
            try{temp=cl.number%temp;}
            catch(ArithmeticException excp)
            {cl.displayLabel.setText("Divide by 0.");
             return;}
            break;
        case '/':
            try{temp=cl.number/temp;}
            catch(ArithmeticException excp)
            {cl.displayLabel.setText("Divide by 0.");
             return;}
            break;
    }

```

ii. Write Implementation

```

MyDigitButton(int x,int y, int width,int height,String cap,
MyCalculator clc)

```

```

{
    super(cap);
    setBounds(x,y,width,height);
    this.cl=clc;
    this.cl.add(this);
    addActionListener(this);
}

```

```

static boolean isInString(String s, char ch)
{
    for(int i=0; i<s.length();i++)
        if(s.charAt(i)==ch)

```

```

        return true;
    }
    return false;
}

public void actionPerformed(ActionEvent ev)
{
    String
tempText=( (MyDigitButton)ev.getSource()).getLabel();

    if(tempText.equals("."))
    {
        if(cl.setClear)
            {cl.displayLabel.setText("0.");cl.setClear=false;}
        else
            if(!isInString(cl.displayLabel.getText(), '.'))

cl.displayLabel.setText(cl.displayLabel.getText()+".");
        return;
    }

    int index=0;

    try{
        index=Integer.parseInt(tempText);
    }catch(NumberFormatException e){
        return;}

    if (index==0 &&
cl.displayLabel.getText().equals("0"))
        return;

    if(cl.setClear)

```

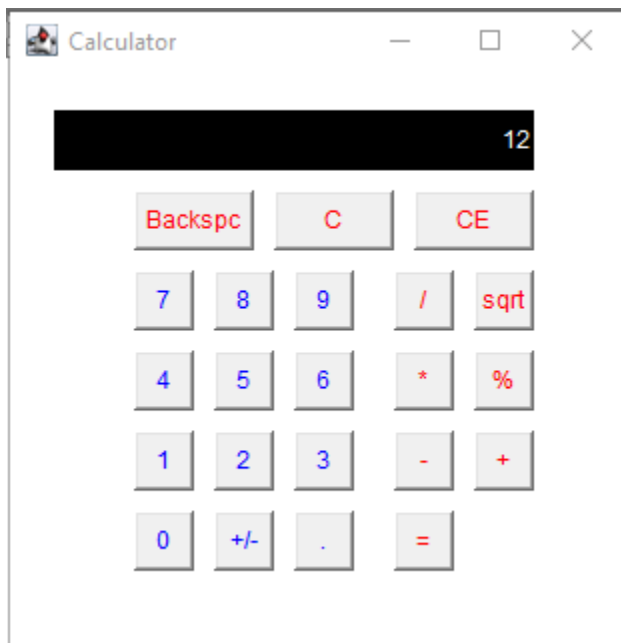
```
{cl.displayLabel.setText(""+index);cl.setClear=false;}  
    else  
  
cl.displayLabel.setText(cl.displayLabel.getText()+index);  
    }  
}
```

5.0 USER MANUAL

i. Interface Calculator.



ii. Choose any number.



iii. Next, click '+' to addition the number. Then choose another number to make the addition.



iv. After that click '=' to get the result.



v. The other buttons is all the same step to use it as it already shown at the above.