ST LAB 4

SAHIL BONDRE: U18CO021

Write a java event handling program to create a scientific calculator.

NOTES :

- Calculator can take input from key board as well.
- Add validations
  - e.g. for fractionl number only one dot is allowed.
- Screenshot given below is for reference GUI .

Calculator.java

```
package lab_4;

import javax.swing.*;
import java.awt.*;
import java.math.BigDecimal;
import java.math.RoundingMode;
import java.util.ArrayList;

public class Calculator {

    private final int WIDTH = 713; // 538
    private final int HEIGHT = 316;
    private final int MARGIN = 10;
    private final int TFHEIGHT = 48;
    private final JFrame window;
    private JTextField tf;
    private ArrayList<JButton> numberBtns;
    private JButton zero, dot, clr, plus, sub, mult, div, sin, cos, tan,
eq, cap, cap2, cap3, fact, open, close, emp, sqrt, log, pi, ee, sinh,
cosh, tanh, exp;
    private JLabel warningLabel;


    public Calculator() {
        window = new JFrame("Scientific Calculator");
        initializeTextField();
```

```
        initializeButtons();

        warningLabel = new JLabel();
        warningLabel.setForeground(Color.RED);
        warningLabel.setBounds(MARGIN, 45, 500, 100);
        window.add(warningLabel);


        window.setSize(WIDTH, HEIGHT);
        window.setLayout(null);
        window.setVisible(true);
        window.setResizable(false);
        window.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    }

    private static double round(double value) {

        BigDecimal bd = BigDecimal.valueOf(value);
        bd = bd.setScale(4, RoundingMode.HALF_UP);
        return bd.doubleValue();
    }

    private void pushCharacter(String character) {
        tf.setText(tf.getText() + character);
    }

    private void popCharacter() {
        String currentString = tf.getText().toString();
        if (currentString.length() >= 1) {
            tf.setText(currentString.substring(0, currentString.length()
- 1));
        }
    }

    private void clearText() {
        tf.setText("");
        warningLabel.setText("");
    }

    private void evaluate() {
        warningLabel.setText("");
        try {
            double res = Evaluator.eval(tf.getText().toString());
            res = round(res);
            tf.setText(Double.toString(res));
```

```
        } catch (Exception e) {
            warningLabel.setText("Error: " + e.getMessage());
        }
    }

    private void initializeButtons() {
        int startX = MARGIN;
        int startY = TFHEIGHT * 2 + MARGIN;
        int rowNum = 0;
        int colNum = 0;
        numberBtns = new ArrayList<>();
        int BTNHEIGHT = 32;
        int BTNWIDTH = 67;
        for (int i = 9; i >= 1; --i) {
            rowNum = ((i - 1) % 3);
            colNum = 3 - ((i - 1) / 3) - 1;
            String val = Integer.toString(i);
            JButton btn = new JButton(val);
            btn.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY
+ colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
            btn.addActionListener(e -> pushCharacter(val));
            numberBtns.add(btn);
            window.add(btn);
        }

        // Last row
        rowNum = 0;
        colNum = 3;
        zero = new JButton("0");
        zero.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        zero.addActionListener(e -> pushCharacter("0"));
        window.add(zero);

        rowNum++;
        dot = new JButton(".");
        dot.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        dot.addActionListener(e -> pushCharacter("."));
        window.add(dot);

        rowNum++;
        clr = new JButton("C");
        clr.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        clr.addActionListener(e -> popCharacter());
```

```
        window.add(clr);

        // add sub div mult
        rowNum = 3;
        colNum = 0;
        plus = new JButton("+");
        plus.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        plus.addActionListener(e -> pushCharacter("+"));
        window.add(plus);

        colNum++;
        sub = new JButton("-");
        sub.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        sub.addActionListener(e -> pushCharacter("-"));
        window.add(sub);

        colNum++;
        mult = new JButton("*");
        mult.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        mult.addActionListener(e -> pushCharacter("*"));
        window.add(mult);

        colNum++;
        div = new JButton("/");
        div.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        div.addActionListener(e -> pushCharacter("/"));
        window.add(div);

        startX = (MARGIN + BTNWIDTH) * 4 + MARGIN;
        JPanel sep = new JPanel();
        sep.setBounds(startX, startY, 1, 4 * (MARGIN + BTNHEIGHT) -
MARGIN);
        sep.setBackground(Color.GRAY);
        window.add(sep);

        startX += MARGIN;
        rowNum = 0;
        colNum = 0;

        // trig
        sin = new JButton("sin");
```

```
        sin.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        sin.addActionListener(e -> pushCharacter("sin("));
        window.add(sin);

        colNum++;
        cos = new JButton("cos");
        cos.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        cos.addActionListener(e -> pushCharacter("cos("));
        window.add(cos);

        colNum++;
        tan = new JButton("tan");
        tan.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        tan.addActionListener(e -> pushCharacter("tan("));
        window.add(tan);

        colNum++;
        eq = new JButton("=");
        eq.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), 2 * BTNWIDTH + MARGIN, BTNHEIGHT);
        eq.addActionListener(e -> evaluate());
        window.add(eq);

        colNum = 0;
        rowNum++;

        cap = new JButton("^");
        cap.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        cap.addActionListener(e -> pushCharacter("^"));
        window.add(cap);

        colNum++;
        cap2 = new JButton("x^2");
        cap2.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        cap2.addActionListener(e -> pushCharacter("^2"));
        window.add(cap2);

        colNum++;
        cap3 = new JButton("x^3");
        cap3.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
```

```
        cap3.addActionListener(e -> pushCharacter("^3"));
        window.add(cap3);


        startX += (MARGIN + BTNWIDTH) * 2;
        rowNum = 0;
        colNum = 0;

        // others
        fact = new JButton("fact");
        fact.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        fact.addActionListener(e -> pushCharacter("fact("));
        window.add(fact);

        colNum++;
        open = new JButton("(");
        open.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        open.addActionListener(e -> pushCharacter("("));
        window.add(open);

        colNum++;
        close = new JButton(")");
        close.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        close.addActionListener(e -> pushCharacter(")"));
        window.add(close);

        colNum++;
        emp = new JButton("CLR");
        emp.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        emp.addActionListener(e -> clearText());
        window.add(emp);

        startX += (MARGIN + BTNWIDTH);
        rowNum = 0;
        colNum = 0;

        sqrt = new JButton("sqrt");
        sqrt.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        sqrt.addActionListener(e -> pushCharacter("sqrt("));
        window.add(sqrt);
```

```
        colNum++;
        log = new JButton("log");
        log.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        log.addActionListener(e -> pushCharacter("log("));
        window.add(log);

        colNum++;
        pi = new JButton("pi");
        pi.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        pi.addActionListener(e -> pushCharacter("3.1415"));
        window.add(pi);

        colNum++;
        ee = new JButton("e");
        ee.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        ee.addActionListener(e -> pushCharacter("2.7183"));
        window.add(ee);


        startX += (MARGIN + BTNWIDTH);
        rowNum = 0;
        colNum = 0;

        sinh = new JButton("sinh");
        sinh.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        sinh.addActionListener(e -> pushCharacter("sinh("));
        window.add(sinh);

        colNum++;
        cosh = new JButton("cosh");
        cosh.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        cosh.addActionListener(e -> pushCharacter("cosh("));
        window.add(cosh);

        colNum++;
        tanh = new JButton("tanh");
        tanh.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        tanh.addActionListener(e -> pushCharacter("tanh("));
        window.add(tanh);
```

```
        colNum++;
        exp = new JButton("exp");
        exp.setBounds(startX + rowNum * (MARGIN + BTNWIDTH), startY +
colNum * (MARGIN + BTNHEIGHT), BTNWIDTH, BTNHEIGHT);
        exp.addActionListener(e -> pushCharacter("exp("));
        window.add(exp);
    }

    private void initializeTextField() {
        tf = new JTextField();
        tf.setBounds(MARGIN, MARGIN, WIDTH - 2 * MARGIN, TFHEIGHT);
        tf.setFont(tf.getFont().deriveFont(Font.BOLD, 48f));
        tf.setHorizontalAlignment(SwingConstants.RIGHT);
        window.add(tf);

    }
}
```

Evaluator.java

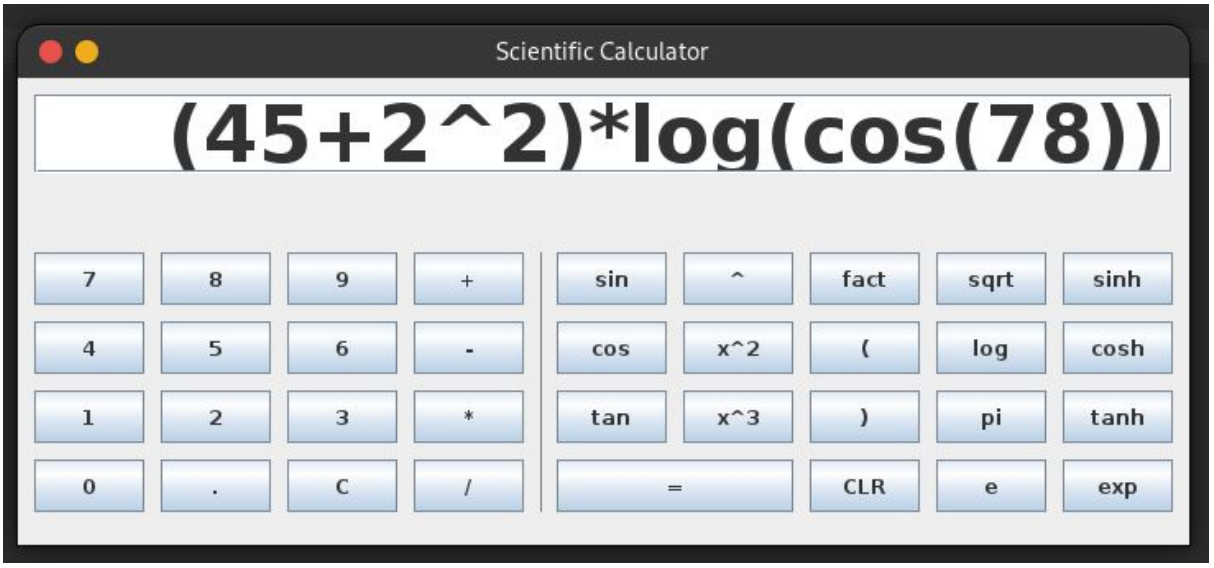
```
package lab_4;

public class Evaluator {
    public static double eval(final String str) {
        return new Object() {
            int pos = -1, ch;

            void nextChar() {
                ch = (++pos < str.length()) ? str.charAt(pos) : -1;
            }

            boolean eat(int charToEat) {
                while (ch == ' ') nextChar();
                if (ch == charToEat) {
                    nextChar();
                    return true;
                }
                return false;
            }

            double parse() {
                nextChar();
                double x = parseExpression();
                if (pos < str.length()) throw new
RuntimeException("Unexpected: " + (char) ch);
                return x;
```

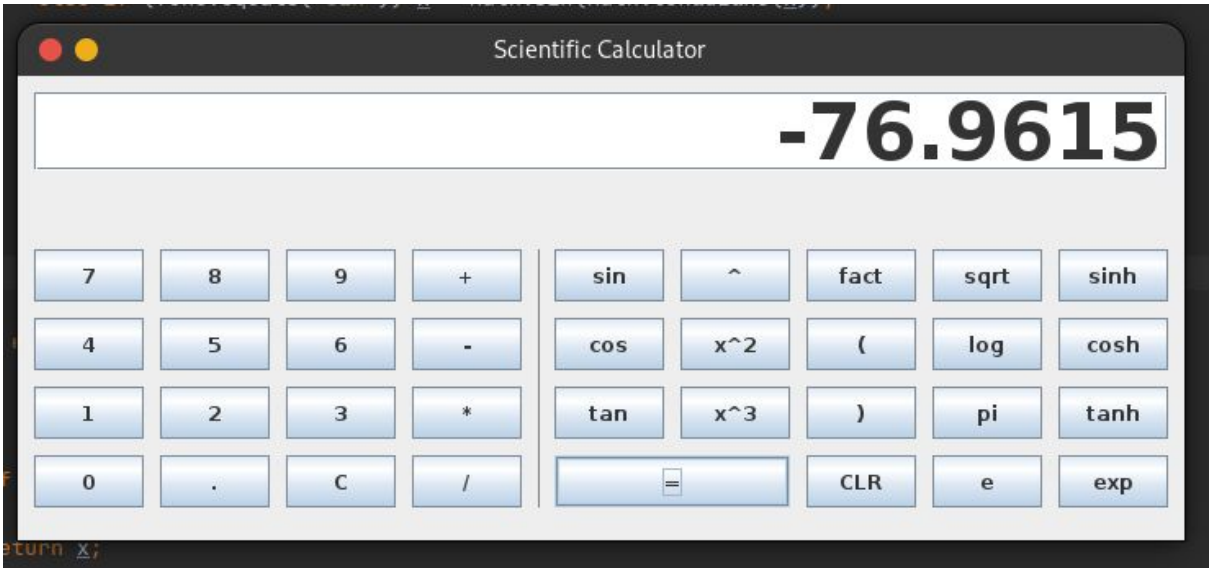
```
            }

            // Grammar:
            // expression = term | expression `+` term | expression `-`
term
            // term = factor | term `*` factor | term `/` factor
            // factor = `+` factor | `-` factor | `(` expression `)`
            //        | number | functionName factor | factor `^` factor

            double parseExpression() {
                double x = parseTerm();
                for (; ; ) {
                    if (eat('+')) x += parseTerm(); // addition
                    else if (eat('-')) x -= parseTerm(); // subtraction
                    else return x;
                }
            }

            double parseTerm() {
                double x = parseFactor();
                for (; ; ) {
                    if (eat('*')) x *= parseFactor(); // multiplication
                    else if (eat('/')) x /= parseFactor(); // division
                    else return x;
                }
            }

            double parseFactor() {
                if (eat('+')) return parseFactor(); // unary plus
                if (eat('-')) return -parseFactor(); // unary minus

                double x;
                int startPos = this.pos;
                if (eat('(')) { // parentheses
                    x = parseExpression();
                    eat(')');
                } else if ((ch >= '0' && ch <= '9') || ch == '.') { //
numbers
                    while ((ch >= '0' && ch <= '9') || ch == '.')
nextChar();
                    x = Double.parseDouble(str.substring(startPos,
this.pos));
                } else if (ch >= 'a' && ch <= 'z') { // functions
                    while (ch >= 'a' && ch <= 'z') nextChar();
                    String func = str.substring(startPos, this.pos);
                    x = parseFactor();
```

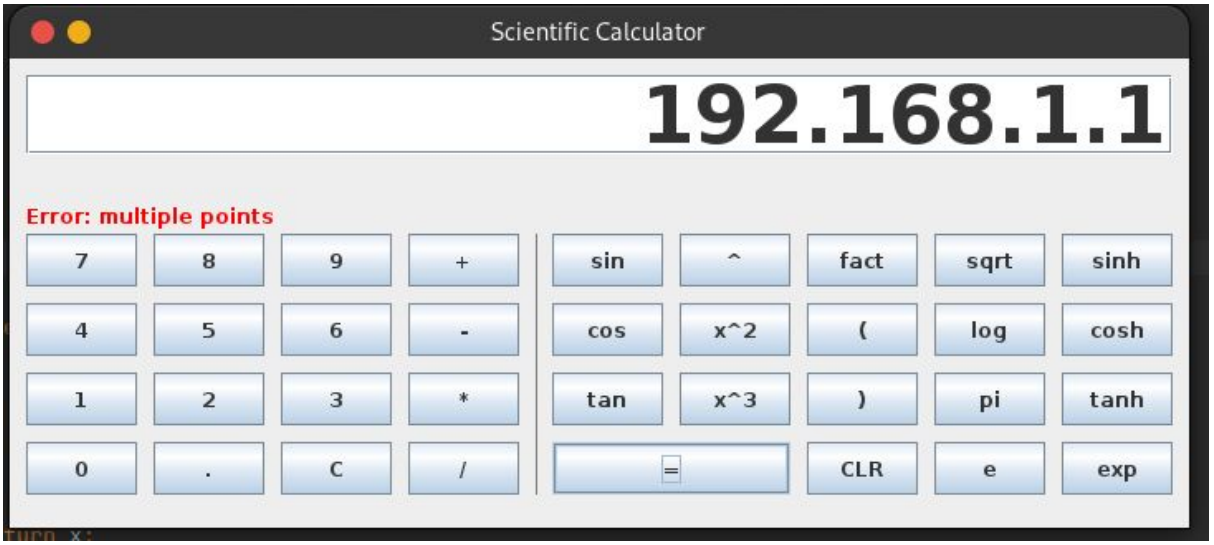
```
                    if (func.equals("sqrt")) x = Math.sqrt(x);
                    else if (func.equals("sin")) x =
Math.sin(Math.toRadians(x));
                    else if (func.equals("cos")) x =
Math.cos(Math.toRadians(x));
                    else if (func.equals("tan")) x =
Math.tan(Math.toRadians(x));
                    else if (func.equals("sinh")) x = Math.sinh(x);
                    else if (func.equals("cosh")) x = Math.sinh(x);
                    else if (func.equals("tanh")) x = Math.sinh(x);
                    else if (func.equals("exp")) x = Math.exp(x);
                    else if (func.equals("fact")) x = factorial(x);
                    else throw new RuntimeException("Unknown function: "
+ func);
                } else {
                    throw new RuntimeException("Unexpected: " + (char)
ch);
                }

                if (eat('^')) x = Math.pow(x, parseFactor()); //
exponentiation

                return x;
            }
        }.parse();
    }

    private static double factorial(double x) {
        if (x <= 1) return 1.0;
        return x * factorial(x - 1);
    }
}
```

Main.java

```
package lab_4;

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


Scientific Calculator
(45+2^2)*log(cos(78))
7 8 9 + sin ^ fact sqrt sinh
4 5 6 - cos x^2 (log cosh
1 2 3 * tan x^3) pi tanh
0 . C / = CLR e exp


Scientific Calculator
-76.9615
7 8 9 + sin ^ fact sqrt sinh
4 5 6 - cos x^2 (log cosh
1 2 3 * tan x^3) pi tanh
0 . C / = CLR e exp


Scientific Calculator
192.168.1.1
Error: multiple points
7 8 9 + sin ^ fact sqrt sinh
4 5 6 - cos x^2 (log cosh
1 2 3 * tan x^3) pi tanh
0 . C / = CLR e exp