

EXPERIMENT:4

User Interface for Calculator

Aim:

To design a user interface by using task analysis for calculator.

Program:

```
import javax.swing.*;

public class NewJFram2 extends JFrame {

    private JPanel jPanel1;
    private JTextField txtResult;
    private JButton[] numberButtons;
    private JButton btnClear, btnPlusMinus, btnPlus, btnMinus, btnMultiply, btnDivide,
    btnEquals;

    private static int value1, value2;
    private static String operator;

    public NewJFram2() {
        initComponents();
    }

    private void initComponents() {
        jPanel1 = new JPanel();
        txtResult = new JTextField();
        numberButtons = new JButton[10];
        for (int i = 0; i < 10; i++) {
            numberButtons[i] = new JButton(String.valueOf(i));
            numberButtons[i].addMouseListener(new java.awt.event.MouseAdapter() {
                public void mouseClicked(java.awt.event.MouseEvent evt) {
                    numberButtonMouseClicked(evt);
                }
            });
        }
        btnClear = new JButton("CE");
        btnPlusMinus = new JButton("/+/-");
        btnPlus = new JButton("+");
        btnMinus = new JButton("-");
        btnMultiply = new JButton("*");
        btnDivide = new JButton("/");
        btnEquals = new JButton("=");

        btnClear.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
```

```

        txtResult.setText("");
    }
});

btnPlus.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        setOperator("plus");
    }
});

btnMinus.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        setOperator("minus");
    }
});

btnMultiply.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        setOperator("multiplication");
    }
});

btnDivide.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        setOperator("division");
    }
});

btnEquals.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        calculateResult();
    }
});

GroupLayout jPanel1Layout = new GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(GroupLayout.Alignment.LEADING)
        .addComponent(txtResult)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(numberButtons[1])
            .addComponent(numberButtons[4])
            .addComponent(numberButtons[7])
            .addComponent(btnPlusMinus))
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(numberButtons[2])
            .addComponent(numberButtons[5])
            .addComponent(numberButtons[8])
            .addComponent(btnDivide))
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(numberButtons[3])
            .addComponent(numberButtons[6])
            .addComponent(numberButtons[9])
            .addComponent(btnMultiply))
        .addComponent(btnEquals));

```

```

        .addComponent(numberButtons[8])
        .addComponent(numberButtons[0]))

    .addGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.LEADING)
        .addComponent(numberButtons[3])
        .addComponent(numberButtons[6])
        .addComponent(numberButtons[9])
        .addComponent(btnClear))

    .addGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.LEADING)
        .addComponent(btnDivide)
        .addComponent(btnMultiply)
        .addComponent(btnMinus)
        .addComponent(btnPlus)))
    .addComponent(btnEquals)
    );

    jPanel1Layout.setVerticalGroup(
        jPanel1Layout.createSequentialGroup()
        .addComponent(txtResult, GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)

    .addGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
        .addComponent(numberButtons[1])
        .addComponent(numberButtons[2])
        .addComponent(numberButtons[3])
        .addComponent(btnDivide))

    .addGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
        .addComponent(numberButtons[4])
        .addComponent(numberButtons[5])
        .addComponent(numberButtons[6])
        .addComponent(btnMultiply))

    .addGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
        .addComponent(numberButtons[7])
        .addComponent(numberButtons[8])
        .addComponent(numberButtons[9])
        .addComponent(btnMinus))

    .addGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
        .addComponent(btnPlusMinus)
        .addComponent(numberButtons[0])
        .addComponent(btnClear)
        .addComponent(btnPlus))
    .addComponent(btnEquals)
    );

    getContentPane().setLayout(new GroupLayout(getContentPane()));
    getContentPane().add(jPanel1, GroupLayout.Alignment.LEADING);

```

```

        pack();
    }

    private void numberButtonMouseClicked(java.awt.event.MouseEvent evt) {
        JButton clickedButton = (JButton) evt.getSource();
        String buttonText = clickedButton.getText();
        if (txtResult.getText().isEmpty()) {
            txtResult.setText(buttonText);
            value1 = Integer.parseInt(buttonText);
        } else {
            txtResult.setText(txtResult.getText() + " " + buttonText);
            value2 = Integer.parseInt(buttonText);
        }
    }

    private void setOperator(String op) {
        if (!txtResult.getText().isEmpty()) {
            operator = op;
            txtResult.setText(txtResult.getText() + " " + getOperatorSymbol(op));
        }
    }

    private String getOperatorSymbol(String op) {
        switch (op) {
            case "plus":
                return "+";
            case "minus":
                return "-";
            case "multiplication":
                return "*";
            case "division":
                return "/";
            default:
                return "";
        }
    }

    private void calculateResult() {
        double result = 0;
        switch (operator) {
            case "plus":
                result = value1 + value2;
                break;
            case "minus":
                result = value1 - value2;
                break;
            case "multiplication":
                result = value1 * value2;
                break;
            case "division":

```

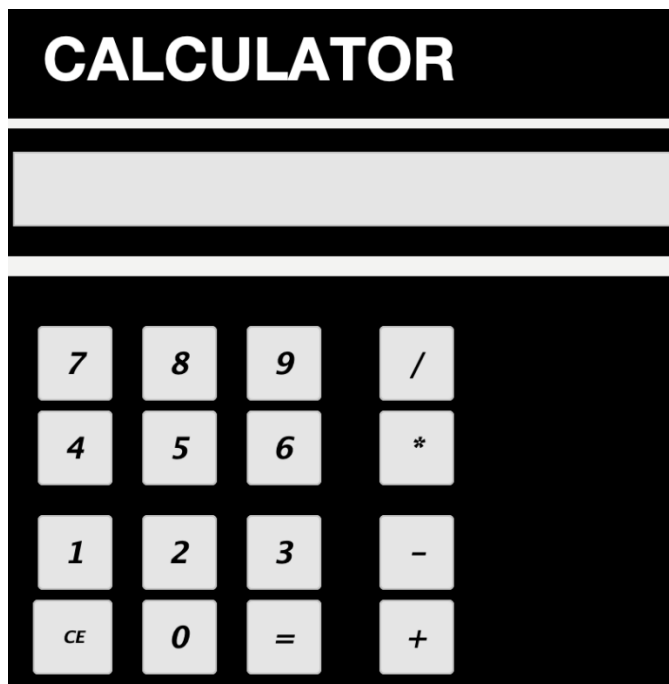
```

        result = value1 / (double) value2;
        break;
    }
    txtResult.setText(Double.toString(result));
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(() -> new NewJFram2().setVisible(true));
}
}

```

Output:



Result:

A user interface by using task analysis for calculator was designed and implemented successfully.