**DESKTOP UTILITY PROJECT**

**(BITWISE CALCULATOR)**

**SOURCE CODE:**

Calculator.java

import java.awt.Color;

import java.awt.Component;

import java.awt.Graphics;

import java.awt.Insets;

import javax.swing.JOptionPane;

import javax.swing.border.Border;

import javax.swing.text.BadLocationException;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

public class Calculator extends javax.swing.JFrame {

    private static final long serialVersionUID = 1L;

    private boolean hasResult = false;

*/\*\**

*\* Creates new form Calculator*

*\*/*

    public Calculator() {

        initComponents();

        addKeyListener(*new* ButtonKeyListener());

        setFocusable(true);

        tf.setForeground(Color.BLACK);

    }

    private void calculate(){

*try* {

*if* (tf.getText().isEmpty()) *return*;

            int result = Bitwise.evaluate(tf.getText());

            tf.setText(String.valueOf(result));

            hasResult = true;

        } *catch* (Exception e) {

            JOptionPane.showMessageDialog(rootPane, "Invalid Expression",

            "Exception", JOptionPane.ERROR\_MESSAGE);

        }

    }

    private class ButtonKeyListener extends KeyAdapter {

        @Override

        public void keyTyped(KeyEvent event) {

*if* (hasResult){

                tf.setText("");

                hasResult = false;

            }

*switch* ((int) event.getKeyChar()) {

*case* 48*:*

                    zeroB.doClick();

*break*;

*case* 49*:*

                    oneB.doClick();

*break*;

*case* 50*:*

                    twoB.doClick();

*break*;

*case* 51*:*

                    threeB.doClick();

*break*;

*case* 52*:*

                    fourB.doClick();

*break*;

*case* 53*:*

                    fiveB.doClick();

*break*;

*case* 54*:*

                    sixB.doClick();

*break*;

*case* 55*:*

                    sevenB.doClick();

*break*;

*case* 56*:*

                    eightB.doClick();

*break*;

*case* 57*:*

                    nineB.doClick();

*break*;

*case* 40*:*

                    open.doClick();

*break*;

*case* 41*:*

                    close.doClick();

*break*;

*case* 38*:*

                    and.doClick();

*break*;

*case* 94*:*

                    exor.doClick();

*break*;

*case* 124*:*

                    or.doClick();

*break*;

*case* 126*:*

                    not.doClick();

*break*;

*case* 8*:*

                    del.doClick();

*break*;

*case* 27*:* *case* 127*:*

                    eraseB.doClick();

*break*;

*case* 61*:* *case* 10*:*

                    equalsB.doClick();

*break*;

*default:*

*break*;

            }

        }

    }

*/\*\**

*\* This method is called from within the constructor to initialize the form.*

*\* WARNING: Do NOT modify this code. The content of this method is always*

*\* regenerated by the Form Editor.*

*\*/*

    @SuppressWarnings("unchecked")

*// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents*

    private void initComponents() {

        java.awt.GridBagConstraints gridBagConstraints;

        jPanel1 = *new* javax.swing.JPanel();

        sevenB = *new* javax.swing.JButton();

        eightB = *new* javax.swing.JButton();

        nineB = *new* javax.swing.JButton();

        fourB = *new* javax.swing.JButton();

        fiveB = *new* javax.swing.JButton();

        sixB = *new* javax.swing.JButton();

        oneB = *new* javax.swing.JButton();

        twoB = *new* javax.swing.JButton();

        threeB = *new* javax.swing.JButton();

        zeroB = *new* javax.swing.JButton();

        eraseB = *new* javax.swing.JButton();

        and = *new* javax.swing.JButton();

        close = *new* javax.swing.JButton();

        or = *new* javax.swing.JButton();

        exor = *new* javax.swing.JButton();

        not = *new* javax.swing.JButton();

        equalsB = *new* javax.swing.JButton();

        del = *new* javax.swing.JButton();

        open = *new* javax.swing.JButton();

        tf = *new* javax.swing.JTextField();

        jMenuBar1 = *new* javax.swing.JMenuBar();

        optionsMenu = *new* javax.swing.JMenu();

        exitbutton = *new* javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

        setTitle("Scientific Calculator");

        setBounds(*new* java.awt.Rectangle(50, 50, 340, 500));

        setUndecorated(true);

        setResizable(false);

        getContentPane().setLayout(*new* java.awt.GridBagLayout());

        jPanel1.setBackground(*new* java.awt.Color(26, 40, 63));

        jPanel1.setForeground(*new* java.awt.Color(26, 40, 63));

        jPanel1.setLayout(null);

        sevenB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/number7.png"))); *// NOI18N*

        sevenB.setBorderPainted(false);

        sevenB.setContentAreaFilled(false);

        sevenB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                sevenBActionPerformed(evt);

            }

        });

        jPanel1.add(sevenB);

        sevenB.setBounds(10, 170, 50, 60);

        eightB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/number8.png"))); *// NOI18N*

        eightB.setBorderPainted(false);

        eightB.setContentAreaFilled(false);

        eightB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                eightBActionPerformed(evt);

            }

        });

        jPanel1.add(eightB);

        eightB.setBounds(60, 170, 50, 60);

        nineB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/number9.png"))); *// NOI18N*

        nineB.setBorderPainted(false);

        nineB.setContentAreaFilled(false);

        nineB.setMaximumSize(*new* java.awt.Dimension(267, 115));

        nineB.setMinimumSize(*new* java.awt.Dimension(267, 115));

        nineB.setPreferredSize(*new* java.awt.Dimension(267, 115));

        nineB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                nineBActionPerformed(evt);

            }

        });

        jPanel1.add(nineB);

        nineB.setBounds(120, 170, 50, 60);

        fourB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/number4.png"))); *// NOI18N*

        fourB.setBorderPainted(false);

        fourB.setContentAreaFilled(false);

        fourB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                fourBActionPerformed(evt);

            }

        });

        jPanel1.add(fourB);

        fourB.setBounds(10, 240, 40, 50);

        fiveB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/number5.png"))); *// NOI18N*

        fiveB.setBorderPainted(false);

        fiveB.setContentAreaFilled(false);

        fiveB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                fiveBActionPerformed(evt);

            }

        });

        jPanel1.add(fiveB);

        fiveB.setBounds(60, 240, 50, 50);

        sixB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/number6.png"))); *// NOI18N*

        sixB.setBorderPainted(false);

        sixB.setContentAreaFilled(false);

        sixB.setMaximumSize(*new* java.awt.Dimension(267, 115));

        sixB.setMinimumSize(*new* java.awt.Dimension(267, 115));

        sixB.setPreferredSize(*new* java.awt.Dimension(267, 115));

        sixB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                sixBActionPerformed(evt);

            }

        });

        jPanel1.add(sixB);

        sixB.setBounds(110, 240, 70, 50);

        oneB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/number1.png"))); *// NOI18N*

        oneB.setBorderPainted(false);

        oneB.setContentAreaFilled(false);

        oneB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                oneBActionPerformed(evt);

            }

        });

        jPanel1.add(oneB);

        oneB.setBounds(10, 300, 50, 50);

        twoB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/number2.png"))); *// NOI18N*

        twoB.setBorderPainted(false);

        twoB.setContentAreaFilled(false);

        twoB.setMaximumSize(*new* java.awt.Dimension(267, 115));

        twoB.setMinimumSize(*new* java.awt.Dimension(267, 115));

        twoB.setPreferredSize(*new* java.awt.Dimension(267, 115));

        twoB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                twoBActionPerformed(evt);

            }

        });

        jPanel1.add(twoB);

        twoB.setBounds(60, 300, 50, 50);

        threeB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/number3.png"))); *// NOI18N*

        threeB.setBorderPainted(false);

        threeB.setContentAreaFilled(false);

        threeB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                threeBActionPerformed(evt);

            }

        });

        jPanel1.add(threeB);

        threeB.setBounds(120, 300, 50, 50);

        zeroB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/number0.png"))); *// NOI18N*

        zeroB.setBorderPainted(false);

        zeroB.setContentAreaFilled(false);

        zeroB.setMaximumSize(*new* java.awt.Dimension(267, 115));

        zeroB.setMinimumSize(*new* java.awt.Dimension(267, 115));

        zeroB.setPreferredSize(*new* java.awt.Dimension(267, 115));

        zeroB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                zeroBActionPerformed(evt);

            }

        });

        jPanel1.add(zeroB);

        zeroB.setBounds(60, 360, 50, 60);

        eraseB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/cancell\_1.png"))); *// NOI18N*

        eraseB.setBorderPainted(false);

        eraseB.setContentAreaFilled(false);

        eraseB.setMaximumSize(*new* java.awt.Dimension(267, 115));

        eraseB.setMinimumSize(*new* java.awt.Dimension(267, 115));

        eraseB.setPreferredSize(*new* java.awt.Dimension(267, 115));

        eraseB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                eraseBActionPerformed(evt);

            }

        });

        jPanel1.add(eraseB);

        eraseB.setBounds(180, 140, 80, 50);

        and.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/and-and.png"))); *// NOI18N*

        and.setBorderPainted(false);

        and.setContentAreaFilled(false);

        and.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                andActionPerformed(evt);

            }

        });

        jPanel1.add(and);

        and.setBounds(180, 250, 80, 70);

        close.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/close.png"))); *// NOI18N*

        close.setBorderPainted(false);

        close.setContentAreaFilled(false);

        close.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                closeActionPerformed(evt);

            }

        });

        jPanel1.add(close);

        close.setBounds(260, 200, 80, 50);

        or.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/or-or.png"))); *// NOI18N*

        or.setBorderPainted(false);

        or.setContentAreaFilled(false);

        or.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                orActionPerformed(evt);

            }

        });

        jPanel1.add(or);

        or.setBounds(260, 250, 80, 70);

        exor.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/exor\_1.png"))); *// NOI18N*

        exor.setBorderPainted(false);

        exor.setContentAreaFilled(false);

        exor.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                exorActionPerformed(evt);

            }

        });

        jPanel1.add(exor);

        exor.setBounds(260, 320, 80, 50);

        not.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/not.png"))); *// NOI18N*

        not.setBorderPainted(false);

        not.setContentAreaFilled(false);

        not.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                notActionPerformed(evt);

            }

        });

        jPanel1.add(not);

        not.setBounds(180, 320, 80, 50);

        equalsB.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/equalsborder.png"))); *// NOI18N*

        equalsB.setBorderPainted(false);

        equalsB.setContentAreaFilled(false);

        equalsB.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                equalsBActionPerformed(evt);

            }

        });

        jPanel1.add(equalsB);

        equalsB.setBounds(190, 370, 140, 80);

        del.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/del.png"))); *// NOI18N*

        del.setBorderPainted(false);

        del.setContentAreaFilled(false);

        del.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                delActionPerformed(evt);

            }

        });

        jPanel1.add(del);

        del.setBounds(260, 140, 80, 50);

        open.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/open.png"))); *// NOI18N*

        open.setBorderPainted(false);

        open.setContentAreaFilled(false);

        open.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                openActionPerformed(evt);

            }

        });

        jPanel1.add(open);

        open.setBounds(180, 200, 80, 50);

        tf.setEditable(false);

        tf.setFont(*new* java.awt.Font("Tahoma", 0, 14)); *// NOI18N*

        tf.setForeground(*new* java.awt.Color(250, 250, 250));

        tf.setHorizontalAlignment(javax.swing.JTextField.RIGHT);

        tf.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

        tf.setOpaque(false);

        tf.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                tfActionPerformed(evt);

            }

        });

        jPanel1.add(tf);

        tf.setBounds(20, 20, 300, 70);

        gridBagConstraints = *new* java.awt.GridBagConstraints();

        gridBagConstraints.gridx = 0;

        gridBagConstraints.gridy = 0;

        gridBagConstraints.ipadx = 339;

        gridBagConstraints.ipady = 466;

        gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;

        getContentPane().add(jPanel1, gridBagConstraints);

        optionsMenu.setIcon(*new* javax.swing.ImageIcon(getClass().getResource("/open-menu.png"))); *// NOI18N*

        optionsMenu.setText("Bitwise Calculator");

        optionsMenu.setIconTextGap(8);

        exitbutton.setText("Exit");

        exitbutton.addActionListener(*new* java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                exitbuttonActionPerformed(evt);

            }

        });

        optionsMenu.add(exitbutton);

        jMenuBar1.add(optionsMenu);

        setJMenuBar(jMenuBar1);

        getAccessibleContext().setAccessibleName("Bitwise Calculator");

        pack();

        setLocationRelativeTo(null);

    }*// </editor-fold>//GEN-END:initComponents*

    private void nineBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_nineBActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "9");

    }*// GEN-LAST:event\_nineBActionPerformed*

    private void sixBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_sixBActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "6");

    }*// GEN-LAST:event\_sixBActionPerformed*

    private void closeActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_closeActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + ")");

    }*// GEN-LAST:event\_closeActionPerformed*

    private void exorActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_exorActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "^");

    }*// GEN-LAST:event\_exorActionPerformed*

    private void fiveBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_fiveBActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "5");

    }*// GEN-LAST:event\_fiveBActionPerformed*

    private void oneBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_oneBActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "1");

    }*// GEN-LAST:event\_oneBActionPerformed*

    private void twoBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_twoBActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "2");

    }*// GEN-LAST:event\_twoBActionPerformed*

    private void threeBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_threeBActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "3");

    }*// GEN-LAST:event\_threeBActionPerformed*

    private void fourBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_fourBActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "4");

    }*// GEN-LAST:event\_fourBActionPerformed*

    private void sevenBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_sevenBActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "7");

    }*// GEN-LAST:event\_sevenBActionPerformed*

    private void eightBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_eightBActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "8");

    }*// GEN-LAST:event\_eightBActionPerformed*

    private void andActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_andActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "&");

    }*// GEN-LAST:event\_andActionPerformed*

    private void notActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_notActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "~");

    }*// GEN-LAST:event\_notActionPerformed*

    private void orActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_orActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "|");

    }*// GEN-LAST:event\_orActionPerformed*

    private void equalsBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_equalsBActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        calculate();

    }*// GEN-LAST:event\_equalsBActionPerformed*

    private void exitbuttonActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_exitbuttonActionPerformed*

*if*(evt.getSource() == exitbutton){

            int response = JOptionPane.showConfirmDialog(

                    rootPane,

                    "Do you want to exit the program?",

                    "Quit",

                    JOptionPane.OK\_CANCEL\_OPTION,

                    JOptionPane.QUESTION\_MESSAGE);

*if*(response == 0){

                 this.dispose();

                 System.exit(0);

            }

        }

    }*// GEN-LAST:event\_exitbuttonActionPerformed*

    private void zeroBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_zeroBActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "0");

    }*// GEN-LAST:event\_zeroBActionPerformed*

    private void eraseBActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_eraseBActionPerformed*

        tf.setText("");

    }*// GEN-LAST:event\_eraseBActionPerformed*

    private void delActionPerformed(java.awt.event.ActionEvent evt) {*// GEN-FIRST:event\_delActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

*try* {

            tf.setText(tf.getText(0, tf.getText().length() - 1));

        } *catch* (BadLocationException e) {

*// Expected Exception: when the user backspace an empty JTextField*

        }

    }*//GEN-LAST:event\_delActionPerformed*

    private void tfActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_tfActionPerformed*

    }*//GEN-LAST:event\_tfActionPerformed*

    private void openActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_openActionPerformed*

*if* (hasResult){

            tf.setText("");

            hasResult = false;

        }

        tf.setText(tf.getText() + "(");

    }*//GEN-LAST:event\_openActionPerformed*

*/\*\**

*\* @param args the command line arguments*

*\*/*

    public static void main(String args[]) {

*/\* Set the Nimbus look and feel \*/*

*//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">*

*/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.*

*\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html*

*\*/*

*try* {

*for* (javax.swing.UIManager.LookAndFeelInfo info *:* javax.swing.UIManager.getInstalledLookAndFeels()) {

*if* ("Nimbus".equals(info.getName())) {

                    javax.swing.UIManager.setLookAndFeel(info.getClassName());

*break*;

                }

            }

        } *catch* (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(Calculator.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } *catch* (InstantiationException ex) {

            java.util.logging.Logger.getLogger(Calculator.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } *catch* (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(Calculator.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } *catch* (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(Calculator.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        }

*//</editor-fold>*

*/\* Create and display the form \*/*

        java.awt.EventQueue.invokeLater(*new* Runnable() {

            @Override

            public void run() {

*new* Calculator().setLocationRelativeTo(null);

*new* Calculator().setVisible(true);

            }

        });

    }

    private static class RoundedBorder implements Border{

        private int radius;

        RoundedBorder(int radius){

            this.radius = radius;

        }

        @Override

        public Insets getBorderInsets(Component c){

*return* *new* Insets(this.radius+1, this.radius+1, this.radius+2, this.radius);

        }

        @Override

        public boolean isBorderOpaque(){

*return* true;

        }

        @Override

        public void paintBorder(Component c, Graphics g, int x, int y, int width, int height){

            g.drawRoundRect(x,y,width-1, height-1, radius, radius);

        }

    }

*// Variables declaration - do not modify//GEN-BEGIN:variables*

    private javax.swing.JButton and;

    private javax.swing.JButton close;

    private javax.swing.JButton del;

    private javax.swing.JButton eightB;

    private javax.swing.JButton equalsB;

    private javax.swing.JButton eraseB;

    private javax.swing.JMenuItem exitbutton;

    private javax.swing.JButton exor;

    private javax.swing.JButton fiveB;

    private javax.swing.JButton fourB;

    private javax.swing.JMenuBar jMenuBar1;

    private javax.swing.JPanel jPanel1;

    private javax.swing.JButton nineB;

    private javax.swing.JButton not;

    private javax.swing.JButton oneB;

    private javax.swing.JButton open;

    private javax.swing.JMenu optionsMenu;

    private javax.swing.JButton or;

    private javax.swing.JButton sevenB;

    private javax.swing.JButton sixB;

    private javax.swing.JTextField tf;

    private javax.swing.JButton threeB;

    private javax.swing.JButton twoB;

    private javax.swing.JButton zeroB;

*// End of variables declaration//GEN-END:variables*

}

Bitwise.java

import java.util.ArrayList;

import java.util.Stack;

public class Bitwise {

    public static int evaluate(String ex) throws Exception {

        String[] infix = tokenize(ex.toCharArray());

*if* (!isBalancedParenthesis(infix)) *throw* *new* Exception();

        String[] postfix = infixToPostfix(infix);

*return* postfixEvaluate(postfix);

    }

    private static int postfixEvaluate(String[] postfix) throws Exception {

        Stack<String> stack = *new* Stack<>();

*for* (int i = 0; i < postfix.length; ++i){

*if* (postfix[i].matches("^[0-9]+$")){

                stack.push(postfix[i]);

            }

*else* *if* (postfix[i].equals("~")){

                int operand = Integer.parseInt(stack.pop());

                int result = ~ operand;

                stack.push(String.valueOf(result));

            }

*else* *if* (postfix[i].matches("[&^|]")){

                int operand2 = Integer.parseInt(stack.pop());

                int operand1 = Integer.parseInt(stack.pop());

                int result = operate(postfix[i],operand1,operand2);

                stack.push(String.valueOf(result));

            }

        }

*if* (stack.size() != 1){

*throw* *new* Exception();

        }

*return* Integer.parseInt(stack.pop());

    }

    private static int operate(String op, int op1, int op2){

*switch* (op) {

*case* "&"*:*

*return* op1 & op2;

*case* "^"*:*

*return* op1 ^ op2;

*case* "|"*:*

*return* op1 | op2;

*default:*

*return* 0;

        }

    }

    private static boolean isBalancedParenthesis(String[] infix) {

        Stack<String> stack = *new* Stack<>();

*for* (int i = 0; i < infix.length; ++i){

*if* (isOpenParenthesis(infix[i])){

                stack.push(infix[i]);

            }

*else* *if* (isClosingParenthesis(infix[i])){

*if* (stack.isEmpty()) *return* false;

                stack.pop();

            }

        }

*return* stack.isEmpty();

    }

    private static String[] tokenize(char[] ex){

        ArrayList<String> token = *new* ArrayList<>();

*for* (int i = 0; i < ex.length; ++i){

*if* (ex[i] >= '0' && ex[i] <= '9'){

                StringBuilder numBuilder = *new* StringBuilder();

*while* (i < ex.length && ex[i] >= '0' && ex[i] <= '9'){

                    numBuilder.append(ex[i++]);

                }

                token.add(numBuilder.toString());

                --i;

            }

*else* {

                token.add(String.valueOf(ex[i]));

            }

        }

*return* token.toArray(*new* String[]{});

    }

    private static String[] infixToPostfix(String[] infix) throws Exception {

        ArrayList<String> postfix = *new* ArrayList<>();

        Stack<String> stack = *new* Stack<>();

*for* (int i = 0; i < infix.length; ++i){

*if* (infix[i].equals("~") && (

                (i+1) == infix.length ||

                !isOperand(infix[i+1]) &&

                !isOpenParenthesis(infix[i+1]))){

*throw* *new* Exception();

            }

*if* (isOperand(infix[i])){

                postfix.add(infix[i]);

            }

*else* *if* (isOperator(infix[i])){

*while* (!stack.isEmpty() &&

                !isOpenParenthesis(stack.peek()) &&

                hasHigherPrecedence(stack.peek(),infix[i])){

                    postfix.add(stack.pop());

                }

                stack.push(infix[i]);

            }

*else* *if* (isOpenParenthesis(infix[i])){

                stack.push(infix[i]);

            }

*else* *if* (isClosingParenthesis(infix[i])){

*while* (!stack.isEmpty() && !isOpenParenthesis(stack.peek())){

                    postfix.add(stack.pop());

                }

                stack.pop();

            }

        }

*while* (!stack.isEmpty()){

            postfix.add(stack.pop());

        }

*return* postfix.toArray(*new* String[]{});

    }

    private static boolean isOperand(String token){

*return* token.matches("^[0-9]+$");

    }

    private static boolean isOperator(String token){

*return* token.matches("[~&^|]");

    }

    private static boolean isOpenParenthesis(String token){

*return* token.equals("(");

    }

    private static boolean isClosingParenthesis(String token){

*return* token.equals(")");

    }

    private static boolean hasHigherPrecedence(String stackOp, String currentOp){

*return* precedence(stackOp) > precedence(currentOp);

    }

    private static int precedence(String operator){

*switch* (operator) {

*case* "~"*:*

*return* 4;

*case* "&"*:*

*return* 3;

*case* "^"*:*

*return* 2;

*case* "|"*:*

*return* 1;

*default:*

*return* 0;

        }

    }

}