

```
// Calcolatrice.java
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class Calcolatrice extends JFrame {

    private DisplayCalcolatrice display;
    private JButton[] btnNumeri;
    private JButton btnSomma, btnSottrazione, btnMoltiplicazione,
btnDivisione;
    private JButton btnUguale, btnCancella, btnMemoria, btnRichiama;
    private JLabel lblContatore;
    private int contatoreClic = 0;
    private double numero1 = 0;
    private double memoria = 0;
    private String operazioneCorrente = "";
    private boolean nuovoNumero = true;

    public Calcolatrice() {
        super("Calcolatrice");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setLocationRelativeTo(null);

        initComponents();
        initPannelli();
        initAscoltatori();

        this.pack();
        this.setVisible(true);
    }

    private void initComponents() {
        display = new DisplayCalcolatrice();
        btnNumeri = new JButton[10];
        for (int i = 0; i < 10; i++) {
            btnNumeri[i] = new JButton(String.valueOf(i));
            btnNumeri[i].setFont(new Font("Arial", Font.BOLD, 16));
        }

        btnSomma = new JButton("+");
        btnSottrazione = new JButton("-");
    }

```

```

        btnMoltiplicazione = new JButton("x");
        btnDivisione = new JButton("÷");
        btnUguale = new JButton("=");
        btnCancella = new JButton("C");
        btnMemoria = new JButton("M+");
        btnRichiama = new JButton("MR");

        lblContatore = new JLabel("Operazioni: 0");
        lblContatore.setHorizontalAlignment(JLabel.CENTER);
    }

    private void initPannelli() {
        Container contentPane = this.getContentPane();
        contentPane.setLayout(new BorderLayout(5, 5));

        // Pannello display (Nord)
        JPanel pnlNord = new JPanel(new BorderLayout());
        pnlNord.add(display, BorderLayout.CENTER);
        pnlNord.add(lblContatore, BorderLayout.SOUTH);

        // Pannello numeri (Centro)
        JPanel pnlCentro = new JPanel(new GridLayout(4, 3, 5, 5));
        for (int i = 1; i <= 9; i++) {
            pnlCentro.add(btnNumeri[i]);
        }
        pnlCentro.add(btnCancella);
        pnlCentro.add(btnNumeri[0]);
        pnlCentro.add(btnUguale);

        // Pannello operazioni (Est)
        JPanel pnlEst = new JPanel(new GridLayout(4, 1, 5, 5));
        pnlEst.add(btnSomma);
        pnlEst.add(btnSottrazione);
        pnlEst.add(btnMoltiplicazione);
        pnlEst.add(btnDivisione);

        // Pannello memoria (Sud)
        JPanel pnlSud = new JPanel(new GridLayout(1, 2, 5, 5));
        pnlSud.add(btnMemoria);
        pnlSud.add(btnRichiama);

        // Aggiunta pannelli al contentPane
        contentPane.add(pnlNord, BorderLayout.NORTH);
        contentPane.add(pnlCentro, BorderLayout.CENTER);
        contentPane.add(pnlEst, BorderLayout.EAST);
        contentPane.add(pnlSud, BorderLayout.SOUTH);

        // Setting bordi
        pnlNord.setBorder(BorderFactory.createEmptyBorder(10, 10, 5, 10));
        pnlCentro.setBorder(BorderFactory.createEmptyBorder(5, 10, 5, 5));
    }

```

```

        pnlEst.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 10));
        pnlSud.setBorder(BorderFactory.createEmptyBorder(5, 10, 10, 10));
    }

    private void initAscoltatori() {
        // Ascoltatore anonimo per i bottoni numerici
        for (int i = 0; i < 10; i++) {
            final int numero = i;
            btnNumeri[i].addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                    if (nuovoNumero) {
                        display.setText("");
                        nuovoNumero = false;
                    }
                    display.aggiungiNumero(numero);
                }
            });
        }

        // Ascoltatore per le operazioni (classe interna)
        AscoltaOperazioni ascoltaOp = new AscoltaOperazioni();
        btnSomma.addActionListener(ascoltaOp);
        btnSottrazione.addActionListener(ascoltaOp);
        btnMoltiplicazione.addActionListener(ascoltaOp);
        btnDivisione.addActionListener(ascoltaOp);
        btnUguale.addActionListener(ascoltaOp);
        btnCancella.addActionListener(ascoltaOp);

        // Ascoltatore per la memoria (classe esterna)
        AscoltaMemoria ascoltaMem = new AscoltaMemoria(this);
        btnMemoria.addActionListener(ascoltaMem);
        btnRichiama.addActionListener(ascoltaMem);
    }

    public void aggiornaContatore() {
        contatoreClic++;
        lblContatore.setText("Operazioni: " + contatoreClic);
    }

    public double getDisplayValue() {
        try {
            return Double.parseDouble(display.getText());
        } catch (NumberFormatException e) {
            return 0;
        }
    }

    public void setDisplayValue(double value) {
        display.setText(String.format("%.2f", value).replace(",", ".", "."));
    }

```

```

        nuovoNumero = true;
    }

    public void setMemoria(double valore) {
        this.memoria = valore;
    }

    public double getMemoria() {
        return this.memoria;
    }

    // Classe interna per gestire le operazioni
    private class AscoltaOperazioni implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            String comando = e.getActionCommand();

            if (comando.equals("C")) {
                display.setText("0");
                numero1 = 0;
                operazioneCorrente = "";
                nuovoNumero = true;
                return;
            }

            if (comando.equals("=")) {
                double numero2 = getDisplayValue();
                double risultato = calcolaRisultato(numero1, numero2,
operazioneCorrente);
                setDisplayValue(risultato);
                operazioneCorrente = "";
                aggiornaContatore();
                return;
            }

            numero1 = getDisplayValue();
            operazioneCorrente = comando;
            nuovoNumero = true;
        }

        private double calcolaRisultato(double num1, double num2, String
operazione) {
            switch (operazione) {
                case "+":
                    return num1 + num2;
                case "-":
                    return num1 - num2;
                case "x":
                    return num1 * num2;
                case "÷":

```

```

        if (num2 == 0) {
            JOptionPane.showMessageDialog(null, "Divisione per
zero non consentita", "Errore", JOptionPane.ERROR_MESSAGE);
            return 0;
        }
        return num1 / num2;
    default:
        return num2;
    }
}
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new Calcolatrice();
        }
    });
}
}

```

// DisplayCalcolatrice.java

```

import java.awt.Color;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.SwingConstants;

public class DisplayCalcolatrice extends JTextField {

    public DisplayCalcolatrice() {
        super("0");
        this.setFont(new Font("Arial", Font.BOLD, 24));
        this.setHorizontalAlignment(SwingConstants.RIGHT);
        this.setEditable(false);
        this.setBackground(new Color(230, 255, 230));
    }

    public void aggiungiNumero(int numero) {
        String testo = this.getText();
        if (testo.equals("0")) {
            this.setText(Integer.toString(numero));
        } else {
            this.setText(testo + numero);
        }
    }
}

```

// AscoltaMemoria.java

```

import java.awt.event.ActionEvent;

```

```
import java.awt.event.ActionListener;

public class AscoltaMemoria implements ActionListener {

    private Calcolatrice calcolatrice;

    public AscoltaMemoria(Calcolatrice calcolatrice) {
        this.calcolatrice = calcolatrice;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand().equals("M+")) {
            double valore = calcolatrice.getDisplayValue();
            calcolatrice.setMemoria(valore);
            calcolatrice.aggiornaContatore();
        } else if (e.getActionCommand().equals("MR")) {
            double memoriaValore = calcolatrice.getMemoria();
            calcolatrice.setDisplayValue(memoriaValore);
        }
    }
}
```