```java
// ContatoreApp.java
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class ContatoreApp extends JFrame {

    private DisplayContatore txtDisplay;
    private JButton btnReset, btnMemorizza, btnRichiama;
    private JButton[] btnIncrementa;
    private JButton[] btnDecrementa;
    private JLabel lblOperazioni, lblTarget;
    private int operazioni, targetValue, memoriaValue;

    public ContatoreApp() {
        super("Contatore Multimediale");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setLocationRelativeTo(null);

        // Inizializzazione variabili
        operazioni = 0;
        targetValue = 100;
        memoriaValue = 0;

        // Inizializzazione componenti
        initComponenti();
        initPannelli();
        initAscoltatori();

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

    private void initComponenti() {
```

```
        // Display
        txtDisplay = new DisplayContatore();

        // Label
        lblOperazioni = new JLabel("Operazioni: 0");
        lblOperazioni.setFont(new Font("Sans-Serif", Font.BOLD, 14));

        lblTarget = new JLabel("Target: " + targetValue);
        lblTarget.setFont(new Font("Sans-Serif", Font.BOLD, 14));

        // Pulsanti incrementa
        String[] incrementi = {"+1", "+5", "+10"};
        btnIncrementa = new JButton[incrementi.length];
        for (int i = 0; i < incrementi.length; i++) {
            btnIncrementa[i] = new JButton(incrementi[i]);
            btnIncrementa[i].setBackground(new Color(144, 238, 144)); //
Light green
        }

        // Pulsanti decrementa
        String[] decrementi = {"-1", "-5", "-10"};
        btnDecrementa = new JButton[decrementi.length];
        for (int i = 0; i < decrementi.length; i++) {
            btnDecrementa[i] = new JButton(decrementi[i]);
            btnDecrementa[i].setBackground(new Color(255, 182, 193)); //
Light pink
        }

        // Altri pulsanti
        btnReset = new JButton("Reset");
        btnReset.setBackground(new Color(255, 99, 71)); // Tomato

        btnMemorizza = new JButton("M+");
        btnMemorizza.setBackground(new Color(135, 206, 250)); // Light sky
blue

        btnRichiama = new JButton("MR");
        btnRichiama.setBackground(new Color(135, 206, 250)); // Light sky
blue
    }

    private void initPannelli() {
        Container contentPane = this.getContentPane();
        contentPane.setLayout(new BorderLayout(10, 10));

        // Pannello Nord - Display e contatori
        JPanel pnlNord = new JPanel(new BorderLayout(5, 5));
        pnlNord.setBorder(BorderFactory.createEmptyBorder(10, 10, 5, 10));

        JPanel pnlInfo = new JPanel(new GridLayout(1, 2, 5, 0));
```

```
        pnlInfo.add(lblOperazioni);
        pnlInfo.add(lblTarget);

        pnlNord.add(txtDisplay, BorderLayout.CENTER);
        pnlNord.add(pnlInfo, BorderLayout.SOUTH);

        // Pannello Centro - Pulsanti operazioni
        JPanel pnlCentro = new JPanel(new GridLayout(2, 3, 5, 5));
        pnlCentro.setBorder(BorderFactory.createEmptyBorder(5, 10, 5, 10));

        for (int i = 0; i < btnIncrementa.length; i++) {
            pnlCentro.add(btnIncrementa[i]);
        }

        for (int i = 0; i < btnDecrementa.length; i++) {
            pnlCentro.add(btnDecrementa[i]);
        }

        // Pannello Sud - Pulsanti funzioni
        JPanel pnlSud = new JPanel(new FlowLayout(FlowLayout.CENTER, 10,
5));
        pnlSud.setBorder(BorderFactory.createEmptyBorder(5, 10, 10, 10));

        pnlSud.add(btnReset);
        pnlSud.add(btnMemorizza);
        pnlSud.add(btnRichiama);

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

    private void initAscoltatori() {
        // Ascoltatore 1: Classe anonima per il pulsante Reset
        btnReset.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                txtDisplay.resetValore();
                operazioni = 0;
                lblOperazioni.setText("Operazioni: " + operazioni);
            }
        });

        // Ascoltatore 2: Classe interna per i pulsanti di
incremento/decremento
        AscoltaOperazioni ascoltaOp = new AscoltaOperazioni();
        for (JButton btn : btnIncrementa) {
            btn.addActionListener(ascoltaOp);
        }
```

```
        for (JButton btn : btnDecrementa) {
            btn.addActionListener(ascoltaOp);
        }

        // Ascoltatore 3: Classe esterna per funzionalità di memoria
        AscoltaPulsanti ascoltaMemoria = new AscoltaPulsanti(this);
        btnMemorizza.addActionListener(ascoltaMemoria);
        btnRichiama.addActionListener(ascoltaMemoria);

        // Aggiunta MouseListener per bloccare/sbloccare i pulsanti
        PulsanteMouseAdapter mouseAdapter = new PulsanteMouseAdapter();
        for (JButton btn : btnIncrementa) {
            btn.addMouseListener(mouseAdapter);
        }

        for (JButton btn : btnDecrementa) {
            btn.addMouseListener(mouseAdapter);
        }
    }

    // Getter e setter
    public DisplayContatore getDisplay() {
        return txtDisplay;
    }

    public void setMemoriaValue(int value) {
        this.memoriaValue = value;
    }

    public int getMemoriaValue() {
        return memoriaValue;
    }

    // Classe interna per ascoltare le operazioni
    private class AscoltaOperazioni implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            JButton source = (JButton) e.getSource();
            String comando = source.getText();

            int valore = txtDisplay.getValore();

            switch(comando) {
                case "+1":
                    valore += 1;
                    break;
                case "+5":
                    valore += 5;
                    break;
```

```
                case "+10":
                    valore += 10;
                    break;
                case "-1":
                    valore -= 1;
                    break;
                case "-5":
                    valore -= 5;
                    break;
                case "-10":
                    valore -= 10;
                    break;
            }

            txtDisplay.setValore(valore);
            operazioni++;
            lblOperazioni.setText("Operazioni: " + operazioni);

            // Controllo se target raggiunto
            if (valore == targetValue) {
                JOptionPane.showMessageDialog(ContatoreApp.this,
                        "Congratulazioni! Hai raggiunto il target " +
targetValue + "!",
                        "Target Raggiunto",
                        JOptionPane.INFORMATION_MESSAGE);
            }
        }
    }

    public static void main(String[] args) {
        new ContatoreApp();
    }
}

// DisplayContatore.java
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import javax.swing.BorderFactory;
import javax.swing.JTextField;
import javax.swing.SwingConstants;

public class DisplayContatore extends JTextField {

    private int valore;

    public DisplayContatore() {
        super("0");
        this.valore = 0;
```

```
        // Impostazioni grafiche
        setFont(new Font("Monospaced", Font.BOLD, 36));
        setHorizontalAlignment(SwingConstants.CENTER);
        setEditable(false);
        setBackground(Color.WHITE);
        setBorder(BorderFactory.createCompoundBorder(
                BorderFactory.createLineBorder(Color.BLACK, 2),
                BorderFactory.createEmptyBorder(5, 10, 5, 10)));

        // Dimensioni
        setPreferredSize(new Dimension(200, 60));
    }

    public void setValore(int valore) {
        this.valore = valore;
        setText(String.valueOf(valore));

        // Cambia colore del testo in base al valore
        if (valore > 0) {
            setForeground(new Color(0, 128, 0)); // Dark green
        } else if (valore < 0) {
            setForeground(new Color(178, 34, 34)); // Firebrick
        } else {
            setForeground(Color.BLACK);
        }
    }

    public int getValore() {
        return valore;
    }

    public void resetValore() {
        setValore(0);
    }
}

// AscoltaPulsanti.java
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;

public class AscoltaPulsanti implements ActionListener {

    private ContatoreApp app;

    public AscoltaPulsanti(ContatoreApp app) {
        this.app = app;
    }

    @Override
```

```
    public void actionPerformed(ActionEvent e) {
        String comando = e.getActionCommand();

        if (comando.equals("M+")) {
            // Memorizza valore corrente
            int valoreCorrente = app.getDisplay().getValore();
            app.setMemoriaValue(valoreCorrente);
        } else if (comando.equals("MR")) {
            // Richiama valore memorizzato
            int valoreMemoriz = app.getMemoriaValue();
            app.getDisplay().setValore(valoreMemoriz);
        }
    }
}

// PulsanteMouseAdapter.java
import java.awt.Color;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JButton;

public class PulsanteMouseAdapter extends MouseAdapter {

    @Override
    public void mouseClicked(MouseEvent e) {
        if (e.getButton() == MouseEvent.BUTTON3) { // Tasto destro
            JButton pulsante = (JButton) e.getComponent();

            if (pulsante.isEnabled()) {
                // Blocca pulsante
                pulsante.setEnabled(false);
                pulsante.setBackground(Color.GRAY);
            } else {
                // Sblocca pulsante
                pulsante.setEnabled(true);

                // Ripristina colore originale in base al tipo di pulsante
                String testo = pulsante.getText();
                if (testo.startsWith("+")) {
                    pulsante.setBackground(new Color(144, 238, 144)); //
Light green
                } else if (testo.startsWith("-")) {
                    pulsante.setBackground(new Color(255, 182, 193)); //
Light pink
                }
            }
        }
    }
}
```