

```
// SimulatoreBancomat.java
import java.awt.BorderLayout;
import java.awt.CardLayout;
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 java.text.DecimalFormat;
import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class SimulatoreBancomat extends JFrame {

    private DisplayBancomat display;
    private TastieraBancomat tastiera;
    private JPanel pnlCard; // Pannello con CardLayout per le diverse
    schermate
    private CardLayout cardLayout;

    private String pinCorrente = "1234"; // PIN di default
    private double saldo = 1000.00; // Saldo iniziale
    private double limitePrelievo = 500.00; // Limite di prelievo
    giornaliero
    private double prelievoGiornaliero = 0.00; // Prelievo effettuato nel
    giorno

    private static final String SCHERMATA_PIN = "PIN";
    private static final String SCHERMATA_MENU = "MENU";
    private static final String SCHERMATA_SALDO = "SALDO";
    private static final String SCHERMATA_PRELIEVO = "PRELIEVO";
    private static final String SCHERMATA_DEPOSITO = "DEPOSITO";
    private static final String SCHERMATA_CAMBIO_PIN = "CAMBIO_PIN";

    public SimulatoreBancomat() {
        super("Simulatore Bancomat");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(400, 600);
        this.setLocationRelativeTo(null);

        initComponents();
    }
}
```

```

        initPannelli();
        initAscoltatori();

        setVisible(true);
    }

    private void initComponents() {
        // Display per visualizzare messaggi e risultati
        display = new DisplayBancomat();

        // Tastiera numerica
        tastiera = new TastieraBancomat();

        // CardLayout per gestire le diverse schermate
        cardLayout = new CardLayout();
        pnlCard = new JPanel(cardLayout);
    }

    private void initPannelli() {
        Container contenitore = this.getContentPane();

        // Pannello superiore con display
        JPanel pnlNord = new JPanel(new BorderLayout());
        pnlNord.add(display, BorderLayout.CENTER);

        // Creazione delle diverse schermate
        creaSchermataPIN();
        creaSchermataMenu();
        creaSchermataSaldo();
        creaSchermataPrelievo();
        creaSchermataDeposito();
        creaSchermataCambioPin();

        // Pannello inferiore con tastiera
        JPanel pnlSud = new JPanel(new BorderLayout());
        pnlSud.add(tastiera, BorderLayout.CENTER);

        // Aggiunta pannelli al contenitore principale
        contenitore.add(pnlNord, BorderLayout.NORTH);
        contenitore.add(pnlCard, BorderLayout.CENTER);
        contenitore.add(pnlSud, BorderLayout.SOUTH);

        // Mostra la schermata iniziale (PIN)
        cardLayout.show(pnlCard, SCHERMATA_PIN);
        display.setText("Inserisci il tuo PIN");
    }

    private void initAscoltatori() {
        // La tastiera è già configurata nel suo costruttore
        // Gli ascoltatori per i pulsanti delle varie schermate sono
    }

```

```
configurati nei rispettivi metodi
```

```
}

// Creazione schermata per l'inserimento del PIN
private void creaSchermataPIN() {
    JPanel pnlPin = new JPanel(new BorderLayout());

    JButton btnConferma = new JButton("Conferma");
    JButton btnCancella = new JButton("Cancella");

    JPanel pnlBottoni = new JPanel(new FlowLayout());
    pnlBottoni.add(btnConferma);
    pnlBottoni.add(btnCancella);

    pnlPin.add(new JLabel("Inserisci il PIN e premi Conferma"),
BorderLayout.NORTH);
    pnlPin.add(pnlBottoni, BorderLayout.SOUTH);

    // Configurazione ascoltatori con classe anonima
    btnConferma.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            verificaPIN();
        }
    });

    btnCancella.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            display.setText("");
        }
    });

    pnlCard.add(pnlPin, SCHERMATA_PIN);
}

// Creazione schermata menu principale
private void creaSchermataMenu() {
    JPanel pnlMenu = new JPanel(new GridLayout(3, 2, 10, 10));
    pnlMenu.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));

    JButton btnSaldo = new JButton("Saldo");
    JButton btnPrelievo = new JButton("Prelievo");
    JButton btnDeposito = new JButton("Deposito");
    JButton btnCambioPin = new JButton("Cambio PIN");
    JButton btnRicevuta = new JButton("Ricevuta");
    JButton btnEsci = new JButton("Esci");

    pnlMenu.add(btnSaldo);
    pnlMenu.add(btnPrelievo);
```

```

        pnlMenu.add(btnDeposito);
        pnlMenu.add(btnCambioPin);
        pnlMenu.add(btnRicevuta);
        pnlMenu.add(btnEsci);

        // Configurazione ascoltatori con classe anonima
        AscoltaOperazioni ascoltaOperazioni = new AscoltaOperazioni();

        btnSaldo.addActionListener(ascoltaOperazioni);
        btnPrelievo.addActionListener(ascoltaOperazioni);
        btnDeposito.addActionListener(ascoltaOperazioni);
        btnCambioPin.addActionListener(ascoltaOperazioni);
        btnRicevuta.addActionListener(ascoltaOperazioni);
        btnEsci.addActionListener(ascoltaOperazioni);

        pnlCard.add(pnlMenu, SCHERMATA_MENU);
    }

    // Creazione schermata saldo
    private void creaSchermataSaldo() {
        JPanel pnlSaldo = new JPanel(new BorderLayout(10, 10));
        pnlSaldo.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));

        JLabel lblSaldo = new JLabel();
        lblSaldo.setFont(new Font("Arial", Font.BOLD, 20));
        lblSaldo.setHorizontalAlignment(JLabel.CENTER);

        JButton btnIndietro = new JButton("Indietro");

        pnlSaldo.add(new JLabel("Il tuo saldo attuale:"),
BorderLayout.NORTH);
        pnlSaldo.add(lblSaldo, BorderLayout.CENTER);
        pnlSaldo.add(btnIndietro, BorderLayout.SOUTH);

        btnIndietro.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                cardLayout.show(pnlCard, SCHERMATA_MENU);
                display.setText("Seleziona un'operazione");
            }
        });

        pnlCard.add(pnlSaldo, SCHERMATA_SALDO);
    }

    // Creazione schermata prelievo
    private void creaSchermataPrelievo() {
        JPanel pnlPrelievo = new JPanel(new GridLayout(5, 1, 10, 10));
        pnlPrelievo.setBorder(BorderFactory.createEmptyBorder(20, 20, 20,
20));
    }

```

```

        JButton btn20 = new JButton("20 €");
        JButton btn50 = new JButton("50 €");
        JButton btn100 = new JButton("100 €");
        JButton btn200 = new JButton("200 €");
        JButton btnIndietro = new JButton("Indietro");

        pnlPrelievo.add(btn20);
        pnlPrelievo.add(btn50);
        pnlPrelievo.add(btn100);
        pnlPrelievo.add(btn200);
        pnlPrelievo.add(btnIndietro);

        // Configurazione ascoltatori con classe interna
        GestorePrelievo gestorePrelievo = new GestorePrelievo();

        btn20.addActionListener(gestorePrelievo);
        btn50.addActionListener(gestorePrelievo);
        btn100.addActionListener(gestorePrelievo);
        btn200.addActionListener(gestorePrelievo);

        btnIndietro.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                cardLayout.show(pnlCard, SCHERMATA_MENU);
                display.setText("Seleziona un'operazione");
            }
        });

        pnlCard.add(pnlPrelievo, SCHERMATA_PRELIEVO);
    }

    // Creazione schermata deposito
    private void creaSchermataDeposito() {
        JPanel pnlDeposito = new JPanel(new BorderLayout(10, 10));
        pnlDeposito.setBorder(BorderFactory.createEmptyBorder(20, 20, 20,
20));

        JLabel lblIstruzioni = new JLabel("Digita l'importo da depositare e
premi Conferma");
        lblIstruzioni.setHorizontalAlignment(JLabel.CENTER);

        JPanel pnlBottoni = new JPanel(new FlowLayout());
        JButton btnConferma = new JButton("Conferma");
        JButton btnCancella = new JButton("Cancella");
        JButton btnIndietro = new JButton("Indietro");

        pnlBottoni.add(btnConferma);
        pnlBottoni.add(btnCancella);
        pnlBottoni.add(btnIndietro);
    }

```

```

pnlDeposito.add(lblIstruzioni, BorderLayout.NORTH);
pnlDeposito.add(pnlBottoni, BorderLayout.SOUTH);

btnConferma.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            double importo = Double.parseDouble(display.getText());
            if (importo <= 0) {
                display.setText("Importo non valido");
            } else {
                saldo += importo;
                display.setText("Deposito di " +
formatImporto(importo) + " effettuato");

JOptionPane.showMessageDialog(SimulatoreBancomat.this,
                                "Deposito effettuato con
successo\nNuovo saldo: " + formatImporto(saldo),
                                "Operazione completata",
JOptionPane.INFORMATION_MESSAGE);
                cardLayout.show(pnlCard, SCHERMATA_MENU);
                display.setText("Seleziona un'operazione");
            }
        } catch (NumberFormatException ex) {
            display.setText("Importo non valido");
        }
    }
});

btnCancella.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        display.setText("");
    }
});

btnIndietro.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        cardLayout.show(pnlCard, SCHERMATA_MENU);
        display.setText("Seleziona un'operazione");
    }
});

pnlCard.add(pnlDeposito, SCHERMATA_DEPOSITO);
}

// Creazione schermata cambio PIN
private void creaSchermataCambioPin() {

```

```

JPanel pnlCambioPin = new JPanel(new BorderLayout(10, 10));
pnlCambioPin.setBorder(BorderFactory.createEmptyBorder(20, 20, 20,
20));

JLabel lblIstruzioni = new JLabel("Inserisci il nuovo PIN (4 cifre)
e premi Conferma");
lblIstruzioni.setHorizontalAlignment(JLabel.CENTER);

JPanel pnlBottoni = new JPanel(new FlowLayout());
JButton btnConferma = new JButton("Conferma");
JButton btnCancella = new JButton("Cancella");
JButton btnIndietro = new JButton("Indietro");

pnlBottoni.add(btnConferma);
pnlBottoni.add(btnCancella);
pnlBottoni.add(btnIndietro);

pnlCambioPin.add(lblIstruzioni, BorderLayout.NORTH);
pnlCambioPin.add(pnlBottoni, BorderLayout.SOUTH);

btnConferma.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String nuovoPin = display.getText();

        if (nuovoPin.length() != 4 || !nuovoPin.matches("\\d+")) {
            display.setText("PIN non valido (deve essere di 4
cifre)");
        } else {
            pinCorrente = nuovoPin;
            display.setText("PIN modificato con successo");
            JOptionPane.showMessageDialog(SimulatoreBancomat.this,
                "PIN modificato con successo",
                "Operazione completata",
JOptionPane.INFORMATION_MESSAGE);
            cardLayout.show(pnlCard, SCHERMATA_MENU);
            display.setText("Seleziona un'operazione");
        }
    }
});

btnCancella.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        display.setText("");
    }
});

btnIndietro.addActionListener(new ActionListener() {
    @Override

```

```

        public void actionPerformed(ActionEvent e) {
            cardLayout.show(pnlCard, SCHERMATA_MENU);
            display.setText("Seleziona un'operazione");
        }
    });

    pnlCard.add(pnlCambioPin, SCHERMATA_CAMBIO_PIN);
}

// Metodo per verificare il PIN inserito
private void verificaPIN() {
    String pinInserito = display.getText();

    if (pinInserito.equals(pinCorrente)) {
        cardLayout.show(pnlCard, SCHERMATA_MENU);
        display.setText("Seleziona un'operazione");
    } else {
        display.setText("PIN errato, riprova");
    }
}

// Formattazione importi
private String formatImporto(double importo) {
    DecimalFormat df = new DecimalFormat("0.00 €");
    return df.format(importo);
}

// Classe interna per gestire gli eventi di prelievo
private class GestorePrelievo implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {
        String comando = e.getActionCommand();
        double importo = Double.parseDouble(comando.replace(" €", ""));

        // Controllo se il prelievo supera il limite giornaliero
        if (prelievoGiornaliero + importo > limitePrelievo) {
            display.setText("Superato limite giornaliero di " +
formatImporto(limitePrelievo));
            JOptionPane.showMessageDialog(SimulatoreBancomat.this,
                "Hai superato il limite giornaliero
di prelievo",
                "Operazione non consentita",
JOptionPane.WARNING_MESSAGE);
            return;
        }

        // Controllo se c'è abbastanza saldo
        if (importo > saldo) {
            display.setText("Saldo insufficiente");

```



```

        JOptionPane.showMessageDialog(SimulatoreBancomat.this,
                                        "Saldo insufficiente per effettuare
questa operazione",
                                        "Operazione non consentita",
JOptionPane.WARNING_MESSAGE);
        return;
    }

    // Esecuzione del prelievo
    saldo -= importo;
    prelievoGiornaliero += importo;

    display.setText("Prelievo di " + formatImporto(importo) + "
effettuato");
    JOptionPane.showMessageDialog(SimulatoreBancomat.this,
                                "Prelievo effettuato con successo\nNuovo
saldo: " + formatImporto(saldo),
                                "Operazione completata",
JOptionPane.INFORMATION_MESSAGE);

    cardLayout.show(pnlCard, SCHERMATA_MENU);
    display.setText("Seleziona un'operazione");
}
}

// Classe esterna per gestire le operazioni principali
public class AscoltaOperazioni implements ActionListener {

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

        switch (comando) {
            case "Saldo":
                // Aggiorna il label prima di mostrare la schermata
                JLabel lblSaldo = (JLabel) ((JPanel)
pnlCard.getComponent(2)).getComponent(1);
                lblSaldo.setText(formatImporto(saldo));

                cardLayout.show(pnlCard, SCHERMATA_SALDO);
                display.setText("Consultazione saldo");
                break;

            case "Prelievo":
                cardLayout.show(pnlCard, SCHERMATA_PRELIEVO);
                display.setText("Seleziona importo da prelevare");
                break;

            case "Deposito":
                cardLayout.show(pnlCard, SCHERMATA_DEPOSITO);

```

```

        display.setText("");
        break;

        case "Cambio PIN":
            cardLayout.show(pnlCard, SCHERMATA_CAMBIO_PIN);
            display.setText("");
            break;

        case "Ricevuta":
            String ricevuta = "RICEVUTA OPERAZIONI\n\n" +
                             "Saldo attuale: " + formatImporto(saldo)
+ "\n" +
                             "Prelievo giornaliero: " +
formatImporto(prelievoGiornaliero) + "\n" +
                             "Limite prelievo: " +
formatImporto(limitePrelievo);

            JOptionPane.showMessageDialog(SimulatoreBancomat.this,
                                         ricevuta,
                                         "Ricevuta",
JOptionPane.INFORMATION_MESSAGE);
            break;

        case "Esci":
            // Reset del display e ritorno alla schermata PIN
            display.setText("Inserisci il tuo PIN");
            cardLayout.show(pnlCard, SCHERMATA_PIN);
            break;
    }
}

// Main per avvio applicazione
public static void main(String[] args) {
    new SimulatoreBancomat();
}

// DisplayBancomat.java
import java.awt.Color;
import java.awt.Font;
import javax.swing.BorderFactory;
import javax.swing.JTextField;

public class DisplayBancomat extends JTextField {

    public DisplayBancomat() {
        super();
        configuraDisplay();
    }
}

```

```

private void configuraDisplay() {
    setFont(new Font("Monospaced", Font.BOLD, 20));
    setHorizontalAlignment(JTextField.CENTER);
    setBackground(new Color(220, 255, 220));
    setEditable(false);
    setBorder(BorderFactory.createCompoundBorder(
        BorderFactory.createLineBorder(Color.BLACK, 2),
        BorderFactory.createEmptyBorder(10, 10, 10, 10)
    ));
    setPreferredSize(new java.awt.Dimension(0, 60));
}

// Override per nascondere il PIN durante l'inserimento
@Override
public void setText(String t) {
    if (t.matches("\\d+") && t.length() <= 4) {
        // Se il testo è composto solo da cifre e di lunghezza <= 4,
        mostriamo asterischi
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < t.length(); i++) {
            sb.append("*");
        }
        super.setText(sb.toString());
    } else {
        // Altrimenti mostriamo il testo normalmente
        super.setText(t);
    }
}

// Metodo per ottenere il testo reale (non mascherato)
@Override
public String getText() {
    if (super.getText().matches("\\*+")) {
        // Se il testo è composto solo da asterischi, restituiamo il PIN
        reale (memorizzato internamente)
        return realText;
    } else {
        return super.getText();
    }
}

private String realText = "";

// Override per tracciare il testo reale
public void setRealText(String t) {
    realText = t;
    setText(t);
}

```

```

// Metodo per aggiungere una cifra
public void addDigit(String digit) {
    realText += digit;
    setText(realText);
}

// Metodo per cancellare l'ultima cifra
public void removeLastDigit() {
    if (realText.length() > 0) {
        realText = realText.substring(0, realText.length() - 1);
        setText(realText);
    }
}

// Metodo per cancellare tutto
public void clear() {
    realText = "";
    super.setText("");
}
}

// TastieraBancomat.java
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JPanel;

public class TastieraBancomat extends JPanel {

    private JButton[] btnNumeri;
    private JButton btnCancella, btnClear;
    private SimulatoreBancomat bancomat;

    public TastieraBancomat() {
        super(new GridLayout(4, 3, 5, 5));

        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));
            btnNumeri[i].setPreferredSize(new Dimension(60, 60));
        }

        btnCancella = new JButton("<");
        btnCancella.setFont(new Font("Arial", Font.BOLD, 16));
        btnCancella.setPreferredSize(new Dimension(60, 60));

```

```

        btnClear = new JButton("C");
        btnClear.setFont(new Font("Arial", Font.BOLD, 16));
        btnClear.setPreferredSize(new Dimension(60, 60));

        // Layout della tastiera (stile tastierino telefonico)
        add(btnNumeri[1]);
        add(btnNumeri[2]);
        add(btnNumeri[3]);
        add(btnNumeri[4]);
        add(btnNumeri[5]);
        add(btnNumeri[6]);
        add(btnNumeri[7]);
        add(btnNumeri[8]);
        add(btnNumeri[9]);
        add(btnCancella);
        add(btnNumeri[0]);
        add(btnClear);

        initAscoltatori();
    }

    public void setBancomat(SimulatoreBancomat bancomat) {
        this.bancomat = bancomat;
    }

    private void initAscoltatori() {
        // Ascoltatore per i tasti numerici (utilizzando una classe interna)
        GestoreTastiera gestoreTastiera = new GestoreTastiera();

        for (int i = 0; i < 10; i++) {
            btnNumeri[i].addActionListener(gestoreTastiera);
        }

        btnCancella.addActionListener(gestoreTastiera);
        btnClear.addActionListener(gestoreTastiera);
    }

    // Classe interna per gestire gli eventi della tastiera
    private class GestoreTastiera implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent e) {
            JButton source = (JButton) e.getSource();
            DisplayBancomat display = (DisplayBancomat)
bancomat.getContentPane().getComponent(0).getComponent(0);

            if (source == btnCancella) {
                display.removeLastDigit();
            } else if (source == btnClear) {
                display.clear();
            }
        }
    }

```

```
    } else {  
        // È un tasto numerico  
        String digit = source.getText();  
        display.addDigit(digit);  
    }  
}  
}
```