```java
// ConvertitoreValute.java
import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.text.DecimalFormat;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class ConvertitoreValute extends JFrame {

    private CampoValuta txtImporto, txtRisultato;
    private JComboBox<String> cmbValutaOrigine, cmbValutaDestinazione;
    private JButton btnInverti, btnReset, btnModificaTassi, btnSalvaTassi,
btnCaricaTassi;
    private TassiCambio tassi;

    private final String[] VALUTE = {"Euro", "Dollaro", "Sterlina", "Yen",
"Franco"};

    public ConvertitoreValute() {
        super("Convertitore di Valute");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setLocationRelativeTo(null);

        tassi = new TassiCambio();

        initComponenti();
        initPannelli();
        initAscoltatori();

        pack();
        setVisible(true);
```

```
        aggiornaCambio();
    }

    private void initComponenti() {
        // Campi di testo
        txtImporto = new CampoValuta();
        txtRisultato = new CampoValuta();
        txtRisultato.setEditable(false);

        // Combo box
        cmbValutaOrigine = new JComboBox<>(VALUTE);
        cmbValutaDestinazione = new JComboBox<>(VALUTE);
        cmbValutaDestinazione.setSelectedIndex(1); // Default: Dollaro

        // Pulsanti
        btnInverti = new JButton("⇄");
        btnReset = new JButton("Reset");
        btnModificaTassi = new JButton("Modifica Tassi");
        btnSalvaTassi = new JButton("Salva Tassi");
        btnCaricaTassi = new JButton("Carica Tassi");
    }

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

        // Pannello input
        JPanel pnlInput = new JPanel(new GridLayout(2, 2, 5, 5));
        pnlInput.add(new JLabel("Importo:"));
        pnlInput.add(txtImporto);
        pnlInput.add(new JLabel("Valuta Origine:"));
        pnlInput.add(cmbValutaOrigine);

        // Pannello centrale
        JPanel pnlCentro = new JPanel(new FlowLayout());
        pnlCentro.add(btnInverti);

        // Pannello output
        JPanel pnlOutput = new JPanel(new GridLayout(2, 2, 5, 5));
        pnlOutput.add(new JLabel("Risultato:"));
        pnlOutput.add(txtRisultato);
        pnlOutput.add(new JLabel("Valuta Destinazione:"));
        pnlOutput.add(cmbValutaDestinazione);

        // Pannello tassi
        JPanel pnlTassi = new JPanel(new FlowLayout());
        pnlTassi.add(btnModificaTassi);
        pnlTassi.add(btnSalvaTassi);
        pnlTassi.add(btnCaricaTassi);

        // Pannello bottoni
```

```
        JPanel pnlBottoni = new JPanel(new FlowLayout(FlowLayout.RIGHT));
        pnlBottoni.add(btnReset);

        // Layout principale
        JPanel pnlMain = new JPanel(new GridLayout(5, 1, 10, 10));
        pnlMain.add(pnlInput);
        pnlMain.add(pnlCentro);
        pnlMain.add(pnlOutput);
        pnlMain.add(pnlTassi);
        pnlMain.add(pnlBottoni);

        contenitore.add(pnlMain, BorderLayout.CENTER);
    }

    private void initAscoltatori() {
        // Ascoltatore con classe anonima per il pulsante reset
        btnReset.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                txtImporto.setText("");
                txtRisultato.setText("");
                cmbValutaOrigine.setSelectedIndex(0);
                cmbValutaDestinazione.setSelectedIndex(1);
            }
        });

        // Ascoltatore con classe esterna per i tassi di cambio
        AscoltaValuta ascoltaValuta = new AscoltaValuta(this);
        btnModificaTassi.addActionListener(ascoltaValuta);
        btnSalvaTassi.addActionListener(ascoltaValuta);
        btnCaricaTassi.addActionListener(ascoltaValuta);

        // Ascoltatore per i campi di testo e combobox (classe interna)
        GestoreConversione gestoreConversione = new GestoreConversione();
        txtImporto.addKeyListener(gestoreConversione);
        cmbValutaOrigine.addActionListener(gestoreConversione);
        cmbValutaDestinazione.addActionListener(gestoreConversione);
        btnInverti.addActionListener(gestoreConversione);
    }

    // Metodo per aggiornare la conversione
    public void aggiornaCambio() {
        try {
            if(txtImporto.getText().isEmpty()) {
                txtRisultato.setText("");
                return;
            }

            double importo = Double.parseDouble(txtImporto.getText());
            String valutaOrigine = (String)
```

```
cmbValutaOrigine.getSelectedItem();
            String valutaDestinazione = (String)
cmbValutaDestinazione.getSelectedItem();

            double tassoCambio = tassi.getTassoCambio(valutaOrigine,
valutaDestinazione);
            double risultato = importo * tassoCambio;

            // Formattazione a due decimali
            DecimalFormat df = new DecimalFormat("#.##");
            txtRisultato.setText(df.format(risultato));
        } catch(NumberFormatException e) {
            txtRisultato.setText("Errore");
        }
    }

    // Getter per TassiCambio
    public TassiCambio getTassi() {
        return tassi;
    }

    // Setter per TassiCambio
    public void setTassi(TassiCambio tassi) {
        this.tassi = tassi;
    }

    // Classe interna per gestire gli eventi di conversione
    private class GestoreConversione extends KeyAdapter implements
ActionListener {

        @Override
        public void keyReleased(KeyEvent e) {
            aggiornaCambio();
        }

        @Override
        public void actionPerformed(ActionEvent e) {
            if(e.getSource() == btnInverti) {
                // Scambia le valute
                int idxOrigine = cmbValutaOrigine.getSelectedIndex();
                int idxDestinazione =
cmbValutaDestinazione.getSelectedIndex();

                cmbValutaOrigine.setSelectedIndex(idxDestinazione);
                cmbValutaDestinazione.setSelectedIndex(idxOrigine);

                // Scambia anche gli importi
                if(!txtImporto.getText().isEmpty() &&
!txtRisultato.getText().isEmpty()) {
                    String temp = txtImporto.getText();
```

```
                    txtImporto.setText(txtRisultato.getText());
                    txtRisultato.setText("");
                    aggiornaCambio();
                }
            } else {
                aggiornaCambio();
            }
        }
    }

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

// CampoValuta.java
import java.awt.Color;
import java.awt.Font;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import javax.swing.JTextField;
import javax.swing.BorderFactory;

public class CampoValuta extends JTextField {

    public CampoValuta() {
        super(10);
        setFont(new Font("Arial", Font.PLAIN, 16));
        setHorizontalAlignment(JTextField.RIGHT);
        setBorder(BorderFactory.createLineBorder(Color.GRAY));

        // Validazione per accettare solo numeri e punto decimale
        addKeyListener(new KeyAdapter() {
            @Override
            public void keyTyped(KeyEvent e) {
                char c = e.getKeyChar();
                if (!Character.isDigit(c) && c != '.' && c !=
KeyEvent.VK_BACK_SPACE) {
                    e.consume();
                }

                // Impedisce più di un punto decimale
                if (c == '.' && getText().contains(".")) {
                    e.consume();
                }
            }
        });
    }
```

```
    // Metodo per impostare il valore
    public void setValore(double valore) {
        setText(String.valueOf(valore));
    }

    // Metodo per ottenere il valore
    public double getValore() {
        try {
            return Double.parseDouble(getText());
        } catch (NumberFormatException e) {
            return 0.0;
        }
    }
}

// TassiCambio.java
import java.io.Serializable;
import java.util.HashMap;
import java.util.Map;

public class TassiCambio implements Serializable {

    private Map<String, Map<String, Double>> tassi;

    public TassiCambio() {
        tassi = new HashMap<>();
        inizializzaTassiDefault();
    }

    private void inizializzaTassiDefault() {
        // Tassi di cambio di default (valori di esempio)
        String[] valute = {"Euro", "Dollaro", "Sterlina", "Yen", "Franco"};

        // Inizializza la mappa per ogni valuta
        for (String valuta : valute) {
            tassi.put(valuta, new HashMap<>());
        }

        // Euro
        tassi.get("Euro").put("Euro", 1.0);
        tassi.get("Euro").put("Dollaro", 1.08);
        tassi.get("Euro").put("Sterlina", 0.85);
        tassi.get("Euro").put("Yen", 160.0);
        tassi.get("Euro").put("Franco", 0.96);

        // Dollaro
        tassi.get("Dollaro").put("Euro", 0.93);
        tassi.get("Dollaro").put("Dollaro", 1.0);
        tassi.get("Dollaro").put("Sterlina", 0.79);
        tassi.get("Dollaro").put("Yen", 148.0);
```

```
        tassi.get("Dollaro").put("Franco", 0.89);

        // Sterlina
        tassi.get("Sterlina").put("Euro", 1.18);
        tassi.get("Sterlina").put("Dollaro", 1.27);
        tassi.get("Sterlina").put("Sterlina", 1.0);
        tassi.get("Sterlina").put("Yen", 188.0);
        tassi.get("Sterlina").put("Franco", 1.13);

        // Yen
        tassi.get("Yen").put("Euro", 0.00625);
        tassi.get("Yen").put("Dollaro", 0.00676);
        tassi.get("Yen").put("Sterlina", 0.00532);
        tassi.get("Yen").put("Yen", 1.0);
        tassi.get("Yen").put("Franco", 0.006);

        // Franco
        tassi.get("Franco").put("Euro", 1.04);
        tassi.get("Franco").put("Dollaro", 1.12);
        tassi.get("Franco").put("Sterlina", 0.88);
        tassi.get("Franco").put("Yen", 166.0);
        tassi.get("Franco").put("Franco", 1.0);
    }

    // Metodo per ottenere il tasso di cambio tra due valute
    public double getTassoCambio(String valutaOrigine, String
valutaDestinazione) {
        return tassi.get(valutaOrigine).get(valutaDestinazione);
    }

    // Metodo per modificare un tasso di cambio
    public void setTassoCambio(String valutaOrigine, String
valutaDestinazione, double tasso) {
        tassi.get(valutaOrigine).put(valutaDestinazione, tasso);
        // Aggiorna automaticamente il tasso inverso
        tassi.get(valutaDestinazione).put(valutaOrigine, 1.0 / tasso);
    }

    // Metodo per ottenere tutte le valute
    public String[] getValute() {
        return tassi.keySet().toArray(new String[0]);
    }
}

// AscoltaValuta.java
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileInputStream;
```

```java
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import javax.swing.JComboBox;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class AscoltaValuta implements ActionListener {

    private ConvertitoreValute convertitore;

    public AscoltaValuta(ConvertitoreValute convertitore) {
        this.convertitore = convertitore;
    }

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

        switch (comando) {
            case "Modifica Tassi":
                modificaTassi();
                break;
            case "Salva Tassi":
                salvaTassi();
                break;
            case "Carica Tassi":
                caricaTassi();
                break;
        }
    }

    // Metodo per modificare i tassi di cambio
    private void modificaTassi() {
        TassiCambio tassi = convertitore.getTassi();
        String[] valute = tassi.getValute();

        // Pannello per il dialogo
        JPanel panel = new JPanel(new GridLayout(2, 2));
        panel.add(new JLabel("Valuta Origine:"));
        JComboBox<String> cmbOrigine = new JComboBox<>(valute);
        panel.add(cmbOrigine);

        panel.add(new JLabel("Valuta Destinazione:"));
        JComboBox<String> cmbDestinazione = new JComboBox<>(valute);
        panel.add(cmbDestinazione);
```

```
        // Aggiunge la selezione della valuta destinazione diversa
dall'origine
        cmbOrigine.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if (cmbOrigine.getSelectedIndex() ==
cmbDestinazione.getSelectedIndex()) {
                    int newIdx = (cmbDestinazione.getSelectedIndex() + 1) %
valute.length;
                    cmbDestinazione.setSelectedIndex(newIdx);
                }
            }
        });

        cmbDestinazione.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if (cmbOrigine.getSelectedIndex() ==
cmbDestinazione.getSelectedIndex()) {
                    int newIdx = (cmbOrigine.getSelectedIndex() + 1) %
valute.length;
                    cmbOrigine.setSelectedIndex(newIdx);
                }
            }
        });

        // Mostra il dialogo per selezionare le valute
        int result = JOptionPane.showConfirmDialog(convertitore, panel,
"Seleziona valute",

JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);

        if (result == JOptionPane.OK_OPTION) {
            String valutaOrigine = (String) cmbOrigine.getSelectedItem();
            String valutaDestinazione = (String)
cmbDestinazione.getSelectedItem();

            // Mostra il tasso attuale
            double tassoAttuale = tassi.getTassoCambio(valutaOrigine,
valutaDestinazione);

            // Chiedi il nuovo tasso
            String input = JOptionPane.showInputDialog(convertitore,
                                "Inserisci il nuovo tasso di cambio da " +
valutaOrigine + " a " + valutaDestinazione,
                                tassoAttuale);

            if (input != null && !input.isEmpty()) {
                try {
```

```
                    double nuovoTasso = Double.parseDouble(input);
                    if (nuovoTasso <= 0) {
                        JOptionPane.showMessageDialog(convertitore, "Il
tasso deve essere positivo",
                                                      "Errore",
JOptionPane.ERROR_MESSAGE);
                    } else {
                        tassi.setTassoCambio(valutaOrigine,
valutaDestinazione, nuovoTasso);
                        convertitore.aggiornaCambio();
                    }
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(convertitore, "Formato non
valido",
                                                  "Errore",
JOptionPane.ERROR_MESSAGE);
                }
            }
        }
    }

    // Metodo per salvare i tassi di cambio
    private void salvaTassi() {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.setDialogTitle("Salva tassi di cambio");

        int result = fileChooser.showSaveDialog(convertitore);
        if (result == JFileChooser.APPROVE_OPTION) {
            File file = fileChooser.getSelectedFile();
            try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(file))) {
                out.writeObject(convertitore.getTassi());
                JOptionPane.showMessageDialog(convertitore, "Tassi salvati
con successo",
                                              "Salvataggio completato",
JOptionPane.INFORMATION_MESSAGE);
            } catch (IOException ex) {
                JOptionPane.showMessageDialog(convertitore, "Errore durante
il salvataggio: " + ex.getMessage(),
                                              "Errore",
JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    // Metodo per caricare i tassi di cambio
    private void caricaTassi() {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.setDialogTitle("Carica tassi di cambio");
```

```
        int result = fileChooser.showOpenDialog(convertitore);
        if (result == JFileChooser.APPROVE_OPTION) {
            File file = fileChooser.getSelectedFile();
            try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(file))) {
                TassiCambio tassi = (TassiCambio) in.readObject();
                convertitore.setTassi(tassi);
                convertitore.aggiornaCambio();
                JOptionPane.showMessageDialog(convertitore, "Tassi caricati
con successo",
                                              "Caricamento completato",
JOptionPane.INFORMATION_MESSAGE);
            } catch (IOException | ClassNotFoundException ex) {
                JOptionPane.showMessageDialog(convertitore, "Errore durante
il caricamento: " + ex.getMessage(),
                                              "Errore",
JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}
```