

## Progetto di un pannello che sia maschera di input

```
/**
 * Maschera.java
 *
 * Maschera classe che crea il pannello consentendo di:
 *     inserire articoli (e relativo prezzo), sia in dollari che in euro
 *     avere sempre sott'occhio il totale, sia in dollari che in euro
 *     poter salvare su file, in qualunque momento, l'elenco degli ordini inseriti.
 *
 * @author classe quarta
 * @version 1.00 2011/4/9
 */
import java.io.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
```

```
public class Maschera extends JPanel {
    JTextField nome, prezzo, totale;
    JTextArea elenco;
    JRadioButton usd, euro;
    ButtonGroup grp;
    JButton save, inserisci;
    Box boxRadio, riga1, riga2, riga3, riga4;
```

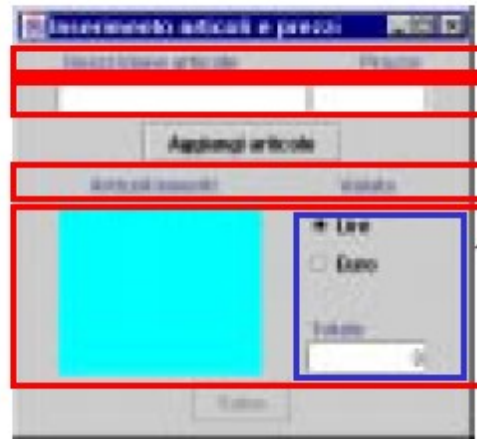
```
// Costruzione del primo Box orizzontale in cui collocare le due etichette
public Maschera(){
    super();
    JLabel etichetta1 = new JLabel("Descrizione articolo");
    JLabel etichetta2 = new JLabel("Prezzo");
    riga1 = new Box(BoxLayout.X_AXIS);
    riga1.add(etichetta1);
    riga1.add(Box.createHorizontalStrut(80)); // spazio fisso indivisibile di 80 pixel
    riga1.add(etichetta2);
```

```
// Costruzione del secondo Box orizzontale in cui collocare i due campi di testo
    nome = new JTextField(15);
    prezzo = new JTextField(7);
    prezzo.setHorizontalAlignment(JTextField.RIGHT);
    riga2 = new Box(BoxLayout.X_AXIS);
    riga2.add(nome);
    riga2.add(prezzo);
```

```
// Costruzione del terzo Box orizzontale con altre etichette
    riga3 = new Box(BoxLayout.X_AXIS);
    riga3.add(new JLabel("Articoli inseriti"));
    riga3.add(Box.createHorizontalStrut(80));
    riga3.add(new JLabel("Valuta"));
```

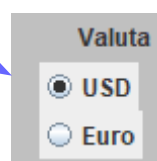
```
//Costruzione del Box verticale per bottoni e totale
// l'area di testo
    elenco = new JTextArea(6,12);
    elenco.setEditable(false); // non modificabile
    elenco.setBackground(Color.cyan);
// i bottoni radio
    usd = new JRadioButton("USD",true); // default
    euro = new JRadioButton("Euro");
    grp = new ButtonGroup();
    grp.add(usd);
    grp.add(euro);
```

### Struttura a blocchi:



Vari Box orizzontali..

.. e un Box verticale.



```
//Costruzione del Box verticale per bottoni e totale
// il Box verticale vero e proprio
boxRadio = new Box(BoxLayout.Y_AXIS);
boxRadio.add(usd); boxRadio.add(euro);
boxRadio.add(Box.createVerticalStrut(20));
boxRadio.add(new JLabel("Totale"));
totale = new JTextField("0", 5); // totale inizialmente a zero
totale.setHorizontalAlignment(JTextField.RIGHT);
boxRadio.add(totale);
```

```
// Costruzione del quarto Box orizzontale per area di testo e Box verticale
riga4 = new Box(BoxLayout.X_AXIS);
riga4.add(elenco);
riga4.add(Box.createHorizontalStrut(30));
riga4.add(boxRadio);
```

```
// Assemblaggio dei vari blocchi
add(riga1);
add(riga2);
inserisci = new JButton("Aggiungi articolo");
add(inserisci);
add(riga3);
add(riga4);
save = new JButton("Salva");
save.setEnabled(false); // inizialmente disabilitato
add(save);
```

```
// Creazione e registrazione dei listener
```

```
inserisci.addActionListener (new InsertListener(nome, prezzo, totale, elenco, usd, save)); // in modo anonimo
CurrencyListener c = new CurrencyListener(prezzo, totale);
usd.addActionListener(c); // unico listener per i due radio-button
euro.addActionListener(c); // unico listener per i due radio-button
save.addActionListener (new SaveListener(elenco, totale, usd, save));
}
} // fine classe che estende JPanel
```

```
class InsertListener implements ActionListener {
```

```
JTextField nome, prezzo, totale;
JTextArea elenco;
JRadioButton usd;
JButton save;
```

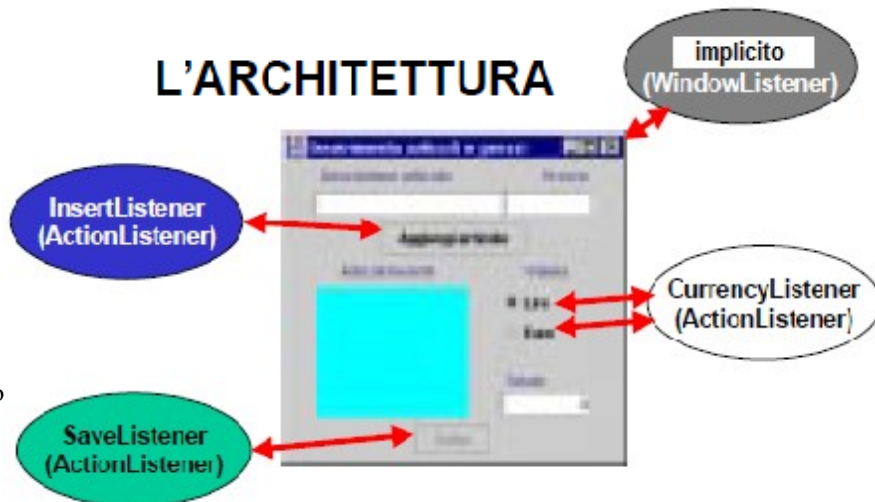
```
public InsertListener (JTextField n, JTextField p, JTextField t, JTextArea e, JRadioButton l, JButton s){
nome=n;
prezzo=p;
totale=t;
elenco=e;
usd=l;
save=s;
}
```

```
public void actionPerformed(ActionEvent e){
save.setEnabled(true);
```

```
double tot = Double.parseDouble(totale.getText());
double prz = Double.parseDouble(prezzo.getText()); // senza controllo degli Input: da gestire con costrutto try – catch
// per evitare NumberFormatException
```

```
totale.setText(Double.toString(prz + tot));
elenco.append (nome.getText() + " " + prezzo.getText() +(usd.isSelected() ? " USD" : " EUR") + "\n");
nome.setText("");
prezzo.setText("");
nome.requestFocus();
}
} // fine classe ascoltatore per inserimento nuovi articoli
```

## L'ARCHITETTURA



## InsertListener

- **Ascolta il pulsante "Aggiungi elemento"**
- **Agisce su tutti i componenti dell'interfaccia**
  - recupera descrizione e prezzo
  - deduce la valuta su cui operare
  - aggiunge l'articolo all'elenco (con prezzo)
  - recupera e aggiorna il totale
  - abilita il pulsante "Salva"

```

class CurrencyListener implements ActionListener {
    JTextField prezzo, totale;
    public final double CAMBIO = 1936.27;
    public CurrencyListener(JTextField p,JTextField t){
        prezzo=p; totale=t;
    }

    public void actionPerformed(ActionEvent e){
        double tot = Double.parseDouble(totale.getText());
        String p = prezzo.getText();
        if (p.equals("")) p="0";
        double price = Double.parseDouble(p);
        if (e.getActionCommand().equals("USD")) {
            // erano Euro, vanno convertiti in dollari
            tot *= CAMBIO;
            price *= CAMBIO;
            // elimina la parte frazionaria di un numero reale,
            tot = Math rint(tot); // arrotondandolo al pari più vicino
            price = Math rint(price);
        }
        else { // erano dollari, da convertire in Euro
            tot /= CAMBIO;
            price /= CAMBIO;
            tot = Math rint(tot*100)/100; // anche con Math.round()
            price = Math rint(price*100)/100;
            // Arrotondamento alla IIª cifra decimale: si sposta la virgola, si arrotonda e si rimette la virgola dov'era
        }
        totale.setText (Double.toString(tot));
        prezzo.setText(price==0.0 ? "" :
            Double.toString(price)); // Estetica: un prezzo di 0.0 non viene indicato, si mette la stringa vuota.
    }
} // fine classe ascoltatore per conversione valuta

```

## CurrencyListener

- Ascolta i due bottoni radio (entrambi!)
- Agisce su prezzo e totale
  - li recupera, converte le stringhe in numeri...
  - ... li moltiplica o divide per il tasso di cambio...
  - ... li riconverte in stringhe e li aggiorna
- Ha il problema di capire quale dei due pulsanti è stato attivato
  - usare getSource è sicuro, ma richiede un riferimento a uno dei due pulsanti
  - usare getActionCommand è più pratico, ma impone modifiche se cambia la label.

```

class SaveListener implements ActionListener {
    JTextField totale;
    JTextArea elenco;
    JRadioButton usd;
    JButton save;
    public SaveListener(JTextArea e, JTextField t, JRadioButton l, JButton s){
        elenco=e;
        totale=t;
        usd=l;
        save=s;
    }

    public void actionPerformed(ActionEvent e){
        double tot = Double.parseDouble(totale.getText());
        try {
            FileWriter f = new FileWriter("ELENCO.txt");
            PrintWriter out = new PrintWriter(f);
            out.print(elenco.getText()); // volendo accodare si dovrebbe gestire il nuovo totale
            out.println();

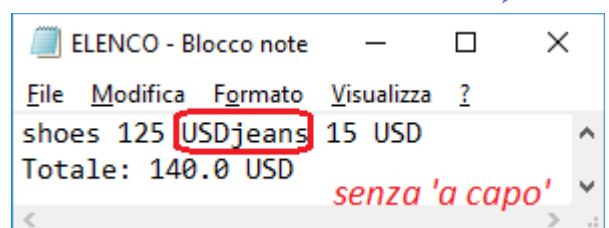
            out.print("Totale: " + totale.getText() +(usd.isSelected() ? " USD" : " EUR") + "\n");
            out.println();
            out.close();
        } catch (IOException ex) { System.err.println(ex); }
        save.setEnabled(false); // disattiva ulteriore salvataggio
    }
} // fine classe ascoltatore per salvataggio su file di testo

```

## SaveListener

- Ascolta il bottone "Salva"
- Agisce su elenco, totale e bottoni radio
  - recupera il testo dell'elenco e lo salva su file
  - salta una riga, recupera il totale e lo salva...
  - ... aggiungendo l'indicazione della valuta.
  - poi disabilita di nuovo il pulsante stesso

// non interpreta 'a capo'  
 // in Blocco note



Di seguito la possibile **classe di test**: applicazione di tipo GUI con pannello personalizzato

```
/**
 * TestMaschera.java
 *
 * TestMaschera application
 *
 * @author quarte
 * @version 1.00 2011/4/15
 */

import java.awt.*;
import javax.swing.*;

class TestMaschera extends JFrame{

    private Container c;
    private Maschera guiP;

    public TestMaschera(){

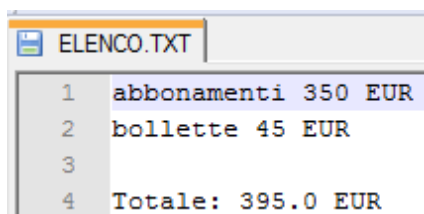
        setTitle("Acquisti");
        c = getContentPane();

        guiP = new Maschera();

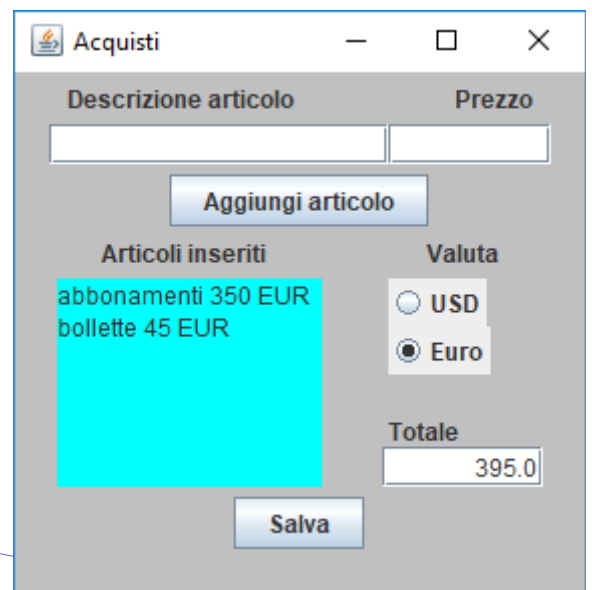
        guiP.setBackground(Color. lightGray);    // sfondo del pannello colorato

        c.add(guiP);           // aggiunge il pannello
        setSize(300,300);
        setVisible(true);
        setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
    }
    public static void main(String [] args){
        new TestMaschera();
    }
}
```

Ad ogni salvataggio, si **riscrive** sul file **senza aggiungere**:

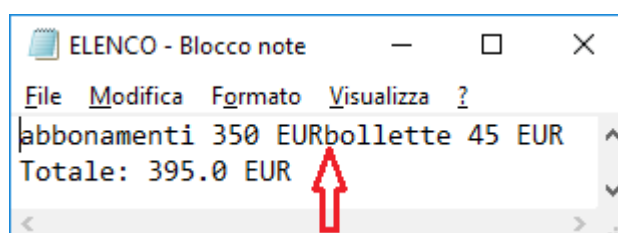


```
ELENCO.TXT
1 abbonamenti 350 EUR
2 bollette 45 EUR
3
4 Totale: 395.0 EUR
```



il file di testo è salvato, alla chiusura dell'applicazione,  
in sottocartella **classes** in JCreator

.... senza gestione dell'*a capo* in Blocco note



```
ELENCO - Blocco note
File Modifica Formato Visualizza ?
abbonamenti 350 EURbollette 45 EUR
Totale: 395.0 EUR
```

Test di *corretta conversione*:

impostato prezzo in USD

... selezionando Euro

The screenshot shows the 'Acquisti' application window. The 'Descrizione articolo' field contains 'prova conversione' and the 'Prezzo' field contains '1.25'. The 'Valuta' section has the 'USD' radio button selected. The 'Articoli inseriti' area is a cyan rectangle. The 'Totale' field shows '0'. The 'Salva' button is visible at the bottom.

The screenshot shows the 'Acquisti' application window. The 'Descrizione articolo' field contains 'prova conversione' and the 'Prezzo' field contains '1.0'. The 'Valuta' section has the 'Euro' radio button selected. The 'Articoli inseriti' area is a cyan rectangle. The 'Totale' field shows '0.0'. The 'Salva' button is visible at the bottom and is highlighted with a blue arrow.

**nota:** se non si aggiunge almeno un articolo, il pulsante per salvare su file è disattivo

### Controllo degli Input

Volendo controllare che l'utente non digiti, la virgola come separatore delle cifre decimali, nell'inserire il prezzo. Di seguito, in caso di tale errore, si propone la forzata uscita dall'applicazione dopo un messaggio di avvertimento

```
// gestione con costrutto try - catch NumberFormatException
// avendo digitato scorrettamente e volendo aggiungere l'articolo

double prz=0;
try{
    prz = Double.parseDouble(prezzo.getText());
} catch(NumberFormatException ne) {
    JOptionPane.showMessageDialog (null, "Separatore punto", "Attenzione!!",
    JOptionPane.ERROR_MESSAGE); // visualizza finestra modale

    System.exit(-1); // esce dall'applicazione
}
```

The screenshot shows the 'Acquisti' application window with the 'biglietto' article and a price of '45,87' (highlighted with a red box). A modal dialog box titled 'Attenzione!!' is displayed in the foreground, containing a red 'X' icon and the text 'Separatore punto'. A red arrow points from the dialog box to the price field in the background window. The 'Salva' button is visible at the bottom.

```

// analogamente
// gestione con costrutto try - catch NumberFormatException
// avendo digitato scorrettamente ed attivando la conversione

double price = 0;
try{
    price = Double.parseDouble(p);
} catch(NumberFormatException ne) {
    JOptionPane.showMessageDialog (null,"Separatore punto", "Attenzione!!",
    JOptionPane.ERROR_MESSAGE); // visualizza finestra modale

    System.exit(-2); // esce dall'applicazione
}

```

