

## I contenitori ScrollPane e TabbedPane

I contenitori di base, come *JFrame* e *JPanel*, consentono di costruire un'area grafica, in cui inserire le componenti dell'interfaccia grafica. La dimensione di questi contenitori di base, e quindi di tutte le loro componenti, può essere al massimo uguale alle dimensioni dello schermo.

Quando l'interfaccia grafica che si vuole progettare richiede che vengano posizionate diverse componenti, la cui dimensione supera l'area dei contenitori di base, si può ricorrere a ulteriori contenitori che consentono di ampliare l'area grafica.

Il contenitore **JScrollPane** gestisce un'area che può essere virtualmente estesa usando i meccanismi di scorrimento verticale e orizzontale. Per esempio, un'immagine di grandi dimensioni, oppure un'immagine a cui viene applicato un effetto di zoom, può essere visualizzata all'interno di un *JScrollPane*. Una normale finestra non avrebbe potuto contenere tutta l'immagine, mentre una finestra con le barre di scorrimento consente all'utente di spostarsi e visualizzare tutto il contenuto.

Il costruttore del *JScrollPane* riceve come parametro la componente di grandi dimensioni che deve essere gestita con le barre di scorrimento. Solitamente questa componente è un *Panel*, su cui vengono posizionati i controlli dell'interfaccia grafica. Se è stato dichiarato un *Panel* di nome *p*, il relativo *JScrollPane* viene dichiarato e creato con la seguente istruzione:

```
JScrollPane scroll = new JScrollPane(p);
```

Per default, le barre di scorrimento appaiono automaticamente quando il pannello supera le dimensioni della finestra grafica, mentre vengono nascoste se il pannello è completamente visibile.

Questo comportamento può essere modificato, imponendo che una o entrambe le barre di scorrimento siano sempre visibili, usando le costanti:

- *JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS*
  - *JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS*
- oppure nascoste, usando le costanti:
- *JScrollPane.VERTICAL\_SCROLLBAR\_NEVER*
  - *JScrollPane.HORIZONTAL\_SCROLLBAR\_NEVER*.

Per esempio, i comandi per impostare la barra verticale sempre visibile e la barra orizzontale nascosta sono:

```
scroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);  
scroll.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_NEVER);
```

## PROGETTO

### Predisporre un'interfaccia grafica per l'inserimento di 30 prodotti.

Questa interfaccia è realizzata usando una tabella avente nella prima colonna le etichette e nella seconda i campi di testo. La tabella viene costruita con un pannello organizzato con un *GridLayout* di 30 righe e due colonne.

Il pannello risulta troppo ampio per essere visualizzato all'interno di una finestra grafica. La soluzione è quella di inserire il pannello in un oggetto *JScrollPane* e posizionare quest'ultimo all'interno della finestra.

### PROGRAMMA JAVA (*ProgScroll.java*)

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

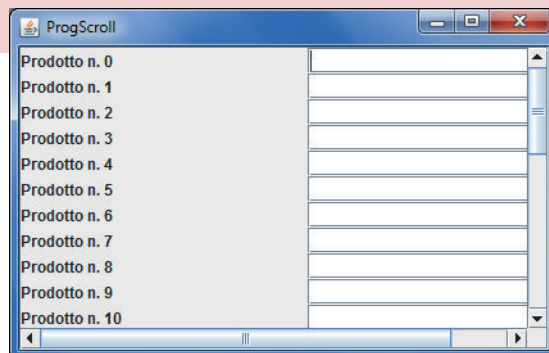
class ProgScroll
{
    public static void main(String argv[])
    {
        JFrame f = new JFrame("ProgScroll");
        JPanel p = new JPanel();

        JTextField articoli[] = new JTextField[30];
        JLabel lbl;

        p.setLayout(new GridLayout(articoli.length,2));
        for(int i=0; i<articoli.length; i++)
        {
            lbl = new JLabel("Prodotto n. " + i);
            articoli[i] = new JTextField(20);
            p.add(lbl);
            p.add(articoli[i]);
        }

        JScrollPane scroll = new JScrollPane(p);

        f.getContentPane().add(scroll);
        f.setSize(450,250);
        f.setLocation(100,100);
        f.setVisible(true);
    }
}
```



Il contenitore **JTabbedPane** permette di condividere lo stesso spazio grafico tra diversi pannelli, detti *tab*, che possono sovrapporsi. L'utente, facendo clic sugli opportuni pulsanti, stabilisce quale dei pannelli deve essere visualizzato.

La costruzione del contenitore *JTabbedPane* prevede la creazione dell'oggetto e, successivamente, l'aggiunta dei pannelli con il metodo **addTab**. Per esempio, un contenitore con due *tab* viene costruito con le seguenti istruzioni:

```
JTabbedPane tabs = new JTabbedPane();

JPanel p1 = new JPanel();
JPanel p2 = new JPanel();

tabs.addTab("Tab 1", p1);
tabs.addTab("Tab 2", p2);
```

Il metodo *addTab* aggiunge in sequenza i pannelli *p1* e *p2* al *JTabbedPane*, assegnando all'etichetta del pulsante del *tab* la stringa indicata nel primo parametro del metodo.

I pulsanti dei *tab* sono solitamente posizionati nella parte superiore dello spazio riservato al *JTabbedPane*. Usando il metodo **setTabPlacement** si può cambiare questa posizione, scegliendola tra uno dei seguenti valori:

- *JTabbedPane.TOP*
- *JTabbedPane.BOTTOM*
- *JTabbedPane.LEFT*
- *JTabbedPane.RIGHT*.

Per esempio, le istruzioni seguenti posizionano i pulsanti dei *tab* nella parte sinistra del *JTabbedPane*:

```
tabs.setTabPlacement(JTabbedPane.LEFT);
```

## PROGETTO

### Predisporre un'interfaccia grafica composta da tre pannelli sovrapponibili.

Questa interfaccia costruisce tre pannelli con un diverso colore di sfondo e li aggiunge in sequenza al *JTabbedPane*. I pulsanti dei *tab* vengono visualizzati lungo il lato sinistro dell'interfaccia.

#### PROGRAMMA JAVA (*ProgTabbed.java*)

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

class ProgTabbed
{
    public static void main(String argv[])
    {
        JFrame f = new JFrame("ProgTabbed");
```

```
// Costruzione dei pannelli

JPanel p1 = new JPanel();
JPanel p2 = new JPanel();
JPanel p3 = new JPanel();
p1.setBackground(Color.red);
p2.setBackground(Color.blue);
p3.setBackground(Color.green);

// Aggiunta dei pannelli al JTabbedPane

JTabbedPane tabs = new JTabbedPane();
tabs.setTabPlacement(JTabbedPane.LEFT);
tabs.addTab("Tab 1", p1);
tabs.addTab("Tab 2", p2);
tabs.addTab("Tab 3", p3);

f.getContentPane().add(tabs);
f.setSize(300,300);
f.setLocation(100,100);
f.setVisible(true);
}
}
```

