

Corso sul linguaggio Java

Modulo JAVA5

A1.3 – Gestori di layout

M. Malatesta A1.3 - Gestori di layout-05

1
27/07/2011

Prerequisiti

- Programmazione base in Java
- Utilizzo di classi e oggetti
- Elementi grafici di un'interfaccia

M. Malatesta A1.3 - Gestori di layout-05

2
27/07/2011

Introduzione

Noti i contenitori e i componenti, è ora il momento di apprendere come questi ultimi possano essere disposti nelle finestre e nei pannelli.

Lo scopo di questa Unità è appunto quello di introdurre apposite classi che consentono di disporre in modo gradevole e razionale gli elementi grafici in un'interfaccia.

M. Malatesta A1.3 - Gestori di layout-05

3
27/07/2011

Il layout dei controlli

La disposizione dei componenti in una interfaccia grafica (**layout**), è consentita da strumenti Java detti **Gestori di interfaccia** (*Layout Manager*, **LM**).

Ogni contenitore ha un **LM di default** che viene utilizzato al momento della visualizzazione.

Il LM di default, volendo, può essere modificato.

Il **LM**:

- ad ogni ridimensionamento della finestra, *riposiziona e ridimensiona le componenti*, mantenendo lo schema scelto inizialmente.
- è un oggetto e pertanto *prima di essere utilizzato va creato*.

M. Malatesta A1.3 - Gestori di layout-05

4
27/07/2011

Il layout dei controlli

La creazione di un oggetto **LM** si esegue con:

```
gestore g = new gestore();
```

dove *gestore* può essere di molti tipi. I più comuni sono:

- **FlowLayout**
- **BorderLayout**
- **GridLayout**

Successivamente, il **LM** va assegnato ad un contenitore con la sintassi

```
container.setLayout (g);
```

dove *container* può essere una finestra o un pannello.

M. Malatesta A1.3 - Gestori di layout-05

5
27/07/2011

Il layout dei controlli

Esempio:

```
...  
Frame f = new Frame();  
BorderLayout g = new BorderLayout ();  
f.setLayout (g);  
....  
oppure  
...  
Frame f = new Frame();  
f.setLayout (new BorderLayout());  
....
```

M. Malatesta A1.3 - Gestori di layout-05

6
27/07/2011

1. Il FlowLayout

```
import java.awt.*;
public class TestFlowLayout
{   public static void main(String[] arg)
    {   Frame f = new Frame ("Finestra con pannello");
        Panel p = new Panel();
        Label info_lbl = new Label ("Premere un pulsante");
        Button next_btn = new Button ("Avanti");
        .....
        p.setBackground (Color.lightGray);
        p.setLayout (new FlowLayout (FlowLayout.CENTER));
        p.add (info_lbl);  p.add (next_btn);
        f.setLocation (200,100);
        f.setSize (400, 400); f.add (p); f.setVisible (true);
    }
} // end class
```

M. Malatesta A1.3 - Gestori di layout-05

7
27/07/2011

1. Il FlowLayout

Il **LM FlowLayout** è il gestore standard associato ai pannelli e nell'esempio mostrato, fa assumere alla finestra l'aspetto seguente.



I controlli sono centrati
rispetto al pannello

M. Malatesta A1.3 - Gestori di layout-05

8
27/07/2011

1. Il FlowLayout

Il metodi tipici del **FlowLayout** sono i seguenti:

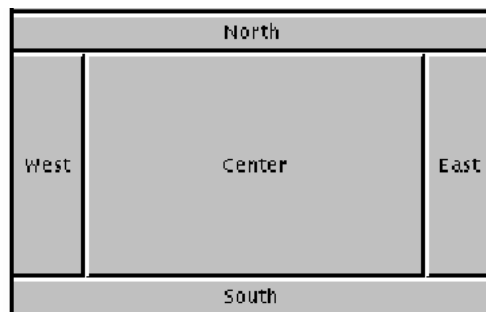
METODO	EFFETTO
<code>FlowLayout ident = new FlowLayout();</code>	Crea un LM di tipo FlowLayout
<code>FlowLayout ident = new FlowLayout(int all);</code>	Crea un LM di tipo FlowLayout con allineamento <i>all</i> (FlowLayout.CENTER , FlowLayout.LEFT , FlowLayout.RIGHT)

M. Malatesta A1.3 - Gestori di layout-05

9
27/07/2011

2. Il BorderLayout

Il LM **BorderLayout** dispone i controlli in 5 aree del contenitore, che hanno nomi come indicato nella figura.



M. Malatesta A1.3 - Gestori di layout-05

10
27/07/2011

2. Il BorderLayout

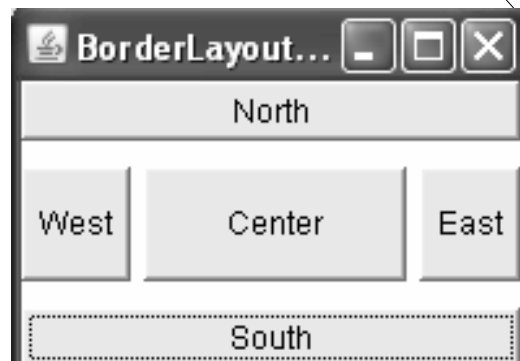
```
import java.awt.*;  
class TestBorderLayout  
{ public static void main(String[] args) {  
    Frame f = new Frame("BorderLayout Example");  
    f.setLayout(new BorderLayout(5, 10));  
    f.add(new Button("North"), BorderLayout.NORTH);  
    f.add(new Button("South"), BorderLayout.SOUTH);  
    f.add(new Button("West"), BorderLayout.WEST);  
    f.add(new Button("East"), BorderLayout.EAST);  
    f.add(new Button("Center"), BorderLayout.CENTER);  
    f.setLocation(100, 100); f.setSize(200, 200); f.setVisible(true);  
}  
}
```

M. Malatesta A1.3 - Gestori di layout-05

11
27/07/2011

2. Il BorderLayout

L'esempio precedente genera la disposizione dei controlli mostrata.



M. Malatesta A1.3 - Gestori di layout-05

12
27/07/2011

2. Il BorderLayout

Il metodi tipici del **BorderLayout** sono i seguenti:

METODO	EFFETTO
<code>BorderLayout ident = new BorderLayout ();</code>	Crea un LM di tipo BorderLayout
<code>BorderLayout ident = new BorderLayout (int hgap, int vgap);</code>	Crea un LM di tipo BorderLayout con spaziatura orizzontale <i>hgap</i> e verticale <i>vgap</i> tra le componenti

M. Malatesta A1.3 - Gestori di layout-05

13
27/07/2011

3. Il GridLayout

Il LM **GridLayout** suddivide l'area di disegno in un reticolato come mostrato in figura. Ogni cella viene acceduta in base alla sequenza numerica riportata.



M. Malatesta A1.3 - Gestori di layout-05

14
27/07/2011

3. Il GridLayout

```
import java.awt.*;  
public class TestGridLayout  
{ public static void main(String[] arg)  
{ Frame f = new Frame("Finestra con pannello");  
  Panel p = new Panel();  
  Label info_lbl = new Label("Premere un pulsante");  
  Button next_btn = new Button ("Avanti");  
  Button prev_btn = new Button ("Indietro");  
  Button first_btn = new Button ("Inizio");  
  Button end_btn = new Button ("Fine");  
  p.setBackground (Color.lightGray);  
  p.setLayout(new GridLayout(2,2, 10,20));  
  p.add(next_btn); p.add(prev_btn); p.add(first_btn); p.add(end_btn);  
  f.setLocation (200,100); f.setSize (400, 400); f.add(p); f.setVisible  
(true);  
}  
} // end class
```

M. Malatesta A1.3 - Gestori di layout-05

15
27/07/2011

3. Il GridLayout

L'esempio precedente genera la disposizione dei controlli mostrata.



M. Malatesta A1.3 - Gestori di layout-05

16
27/07/2011

3. Il GridLayout

Il metodi tipici del **GridLayout** sono i seguenti:

METODO	EFFETTO
GridLayout <i>ident</i> = new GridLayout ();	Crea un LM di tipo GridLayout con un componente per colonna
GridLayout <i>ident</i> = new GridLayout (<i>int</i> <i>r</i> , <i>int</i> <i>c</i>);	Crea un LM di tipo GridLayout con <i>r</i> righe e <i>c</i> colonne
GridLayout <i>ident</i> = new GridLayout (<i>int</i> <i>r</i> , <i>int</i> <i>c</i> , <i>int</i> <i>hgap</i> , <i>int</i> <i>vgap</i>);	Crea un LM di tipo BorderLayout con <i>r</i> righe e <i>c</i> colonne e spaziatura orizzontale <i>hgap</i> e verticale <i>vgap</i> tra le componenti

M. Malatesta A1.3 - Gestori di layout-05

17
27/07/2011

4. Il layout manuale

La disposizione degli elementi grafici sull'interfaccia, può anche essere eseguita "manualmente" tramite il metodo **setBound()** che ha la seguente sintassi:

setBounds (*x*, *y*, *larghezza*, *altezza*)

dove

- *x* ed *y* sono le coordinate dell'angolo superiore sinistro dell'oggetto
- *larghezza* e *altezza* rappresentano, in pixel, le dimensioni dell'oggetto

Per poter utilizzare il posizionamento manuale indicato, è necessario (visto che i contenitori hanno **LM** di default) impostare a **null** il layout predefinito, con l'istruzione

setLayout (**null**);

M. Malatesta A1.3 - Gestori di layout-05

18
27/07/2011

Combinazione di gestori

SLIDE NASCOSTA

Disegno CN 149

M. Malatesta A1.3 - Gestori di layout-05

19
27/07/2011

Argomenti

- Il layout dei controlli
- 1. Il FlowLayout
- 2. Il BorderLayout
- 3. Il GridLayout
- 4. Il layout manuale

M. Malatesta A1.3 - Gestori di layout-05

20
27/07/2011

Altre fonti di informazione

- P.Gallo, F.Salerno – Informatica Generale 1, ed. Minerva Italica
- M.Romagnoli, P.Ventura – Linguaggio C/C++, ed. Pettrini
- M. Bigatti – Il linguaggio Java, ed. Hoepli