

Java GUI

Librerie Awt e Swing

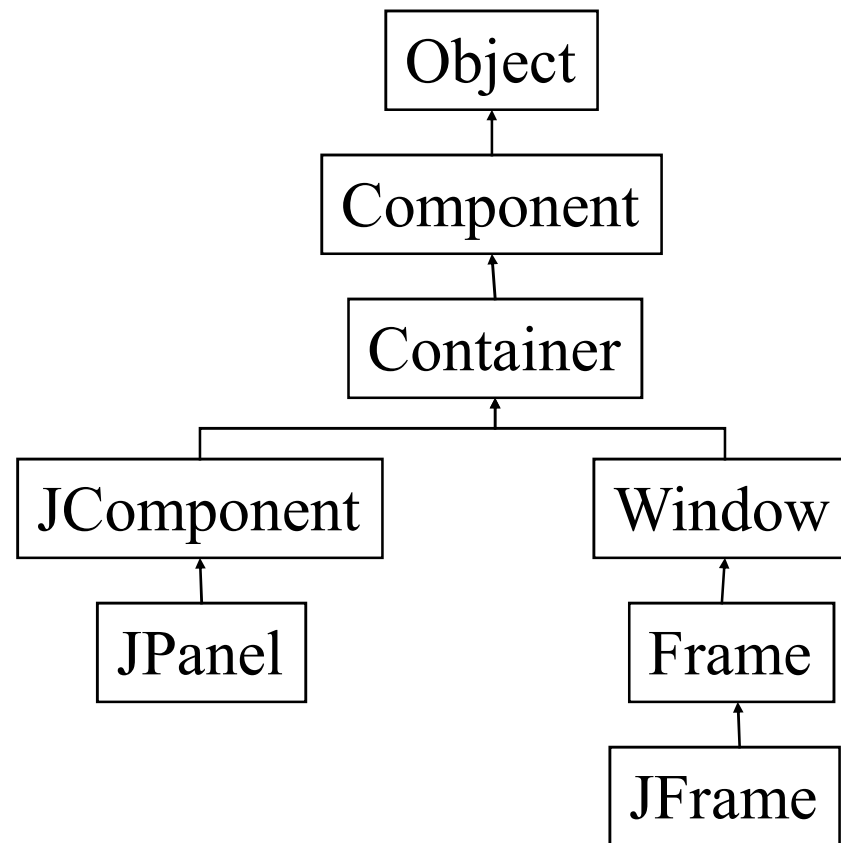
Claudio Borgogno

Contenitori e componenti

- L'interfaccia grafica utente è formata da contenitori e da componenti
 - un contenitore è un frame(finestra) o un pannello
- Ogni contenitore può contenere più componenti
- Ogni contenitore (insieme ai suoi componenti) può essere considerato un unico componente ed essere inserito in altri contenitori
- Le classi per creare interfacce appartengono ai package `java.awt` e `javax.swing` (iniziano con J):
 - JButton, JTextBox, JTextArea, JPanel, JFrame, JMenu, JSlider, JLabel, Icon, ...

Componenti Swing

- Ogni componente è una classe java, con una propria gerarchia:



Usare Componenti Swing

- In modo semplice, si possono creare componenti dalle relative classi:
 - `JButton but = new JButton();`
 - `TextField text = new TextField();`
 - `JTextArea text = new JTextArea();`
 - `JLabel lab = new JLabel();`

Aggiungere componenti

- Una volta che un componente è stato creato, esso può essere aggiunto, attraverso la chiamata del metodo `add()` sul contenitore:

```
Container cp = getContentPane();
```

```
cp.add(new JButton("cancel"));
```

```
cp.add(new JButton("go"));
```

Il modo con cui i componenti vengono sistemati è determinato dal layout manager.

Frame

- Le operazioni da svolgere per creare un frame:
 - creare un'istanza della classe JFrame
 - stabilire la dimensione del frame
 - stabilire la posizione iniziale del frame
 - impostare l'operazione associata al pulsante di chiusura
 - rendere visibile il frame

Creare un'istanza della classe JFrame

- Il costruttore della classe JFrame permette di creare un frame con titolo:

```
public JFrame(String titolo)
```

Dobbiamo pertanto scrivere:

```
JFrame myFrame = new JFrame("Primo");
```

Stabilire la dimensione

- Utilizziamo il metodo `setSize`, ereditato dalla classe `java.awt.Component`:

```
public setSize(int larghezza, int altezza)
```

Dobbiamo pertanto scrivere:

```
myFrame.setSize(WIDTH, HEIGHT);
```


Stabilire la posizione iniziale

- Utilizziamo il metodo setLocation, ereditato dalla classe java.awt.Component:

```
public void setLocation(int ascissa, int ordinata)
```

Dobbiamo pertanto scrivere:

```
myFrame.setLocation(XPOS, YPOS);
```

Stabilire la dimensione e posizione contemporaneamente

- Utilizziamo il metodo `setBounds`, ereditato dalla classe `java.awt.Component`:

```
public void setBounds(int ascissa, int ordinata,  
                      int larghezza, int altezza)
```

Dobbiamo pertanto scrivere:

```
myFrame .setBounds(XPOS, YPOS,  
                   WIDTH, HEIGHT);
```

Determinare automaticamente la dimensione

- Utilizziamo il metodo `pack`, ereditato dalla classe `java.awt.Component`:

```
public void pack()
```

determina automaticamente della finestra sulla base dei suoi componenti.

Dobbiamo pertanto scrivere:

```
myFrame .pack();
```

Impostare l'azione associata al pulsante di chiusura (1)

- Si utilizza il metodo `setDefaultCloseOperation` dalla classe `javax.swing.JFrame`:

```
public void setDefaultCloseOperation()
```

Specifica l'azione da eseguire quando si fa click sul pulsante di chiusura della finestra.

Impostare l'azione associata al pulsante di chiusura (2)

- Il parametro azione può assumere uno dei seguenti valori:
- `JFrame.EXIT_ON_CLOSE`: chiude la finestra e termina l'esecuzione dell'applicazione
- `WindowsConstants.DISPOSE_ON_CLOSE`: chiude la finestra, liberando le risorse di sistema, ma senza interrompere l'esecuzione.
- `WindowsConstants.DO_NOTHING_ON_CLOSE`: non esegue alcuna azione.
- `WindowsConstants.HIDE_ON_CLOSE`: nasconde la finestra (è il valore di default).

Impostare l'azione associata al pulsante di chiusura (3)

- Per terminare l'applicazione alla chiusura del frame, dovremo scrivere:

```
myFrame.setDefaultCloseOperation(  
    JFrame.EXIT_ON_CLOSE)
```

Rendere visibile il frame.

- Si utilizza il metodo setVisible della classe `java.awt.Window`:

```
public void setVisible(boolean b)
```

Se `b=true`, visualizza il componente, altrimenti lo nasconde

Dobbiamo quindi scrivere:

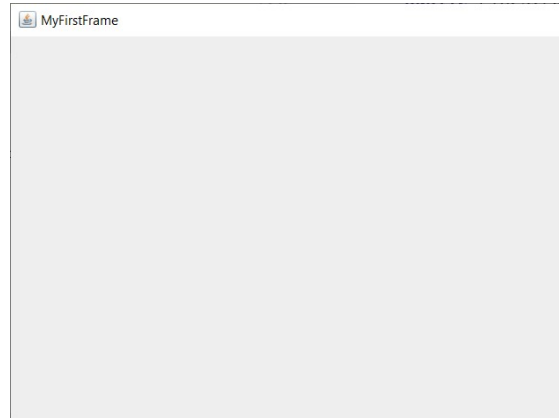
```
myFrame.setVisible(true);
```

Generazione Frame nel main()

- La classe del Frame viene istanziata all'interno del metodo main():

```
public class Finestra{  
  
    public Finestra()  
    {  
    }  
  
    public static void main(String[] args)  
    {  
        JFrame myFrame = new JFrame("Primo");  
        setTitle("Primo");  
        setSize(800,600);  
        setLocation(200, 200);  
        //initComponets();  
        //pack();  
        //setVisible(true);  
    }  
}
```


Output del Frame



Il frame creato ha le seguenti caratteristiche:

- è ridimensionabile mediante il primo pulsante presente sulla barra del titolo
- è riducibile a icona mediante il secondo pulsante presente sulla barra del titolo
- è richiudibile mediante il pulsante X presente sulla barra del titolo
- è dimensionabile con il mouse, agendo sul contorno

Generazione Frame extends JFrame (1)

- E' possibile creare un frame estendendo la classe JFrame
- Le operazioni viste in precedenza devono essere spostate nel costruttore
- La classe può contenere il metodo main che
 - creerà l'istanza del frame

Generazione Frame extends JFrame (2)

- La classe del Frame, all'interno del costruttore Finestra, che estende JFrame:

```
public class Finestra extends JFrame{  
  
    public Finestra()  
    {  
        setTitle("Primo");  
        setSize(800,600);  
        setLocation(200, 200);  
        //initComponets();  
        //pack();  
        setVisible(true);  
        this.setDefaultCloseOperation(DISPOSE  
    }  
  
    public static void main(String[] args)  
    {
```

Osservazioni sulla creazione di un frame

- Se non specifichiamo la dimensione, il frame avrà dimensione nulla
- Se non specifichiamo la posizione iniziale del frame, questa coinciderà con il punto di coordinate (0,0)
- Un frame appena creato, non è visibile.
- Se applichiamo il metodo `pack()`, ad una finestra che non ha componenti, questa ha dimensione nulla, occorre quindi impostarla esplicitamente con `setSize(800, 600);`

Pannelli

Creare un pannello

- Un pannello è usato principalmente per raggruppare i componenti (inclusi altri pannelli) da inserire in un altro contenitore (ad esempio un frame)
 - per tale motivo, un pannello è al tempo stesso sia un contenitore che un componente
- Un pannello viene creato come istanza della classe Panel.
 - `Jpanel myPanel = new JPanel();`

Aggiungere il pannello al frame (1)

- Le operazioni da compiere:
 - ottenere il riferimento al pannello del contenuto del frame
 - Aggiungere il pannello al pannello del contenuto
- Il pannello del contenuto (***contentPane***) è un pannello predefinito contenuto in ogni frame
- Il metodo `getContentPane()` della classe `JFrame`

public Container getContentPane()

consente di ottenere il riferimento al `contentPane`:

```
Container contentPane = myFrame.getContentPane();
```

Aggiungere il pannello al frame (2)

- L'aggiunta del pannello al frame avviene utilizzando il metodo della classe `java.awt.Container`

`public Component add(Component comp)`

inserisce nel contenitore il componente (comp)

Per aggiungere il pannello `myPanel` al pannello del contenuto `myFrame`, si deve scrivere:

```
contentPane = myFrame.getContentPane();
```

```
contentPane.add(myPanel);
```

- Il pannello aggiunto occupa tutto lo spazio interno al frame e non è visibile, a meno che non si imposti un colore per lo sfondo.

Impostare il colore di sfondo di un pannello

- Si utilizza il metodo setBackground della classe Jcomponent:

public void setBackground(Color bg)

- imposta il colore di sfondo del componente a quello specificato; se il parametro è null il componente eredita il colore di sfondo del componente che lo contiene (padre)
- Per impostare il colore di sfondo del pannello myPanel a rosso:
 - myPanel.setBackground(Color.RED);

La classe Color

- La classe Color definisce una serie di costanti utilizzabili immediatamente
- E' possibile creare nuovi colori, utilizzando uno dei costruttori della classe Color:

`Color myColor = new Color(RED, GREEN, BLUE)`

dove le componenti di colore rgb, variano da 0 a 255

`Color myColor = new Color(200, 140, 42);`

Frame con pannello colorato

```
public class Finestra extends JFrame {  
    public Finestra()  
    {  
        setSize(800,600);  
        setLocation(200, 200);  
        setTitle("MyFirstFrame");  
        //initComponets();  
        //pack();  
        JPanel myPanel = new JPanel();  
        myPanel.setBackground(Color.BLUE);  
        Container container = this.getContentPane();  
        container.add(myPanel);  
        setVisible(true);  
        this.setDefaultCloseOperation(DISPOSE_  
    }  
  
    public static void main(String[] args)
```

Componenti

Componenti di base

- I componenti dell'interfaccia utente sono gli elementi che possono essere aggiunti all'interno di contenitori (frame e pannelli).
- I componenti di base sono:
 - etichette (JLabel)
 - caselle di testo (JTextField)
 - pulsanti (Jbutton)

Etichette (1)

- Un etichetta contiene una riga di testo che non può essere modificata dall'utente, ma solo dall'applicazione
- Le etichette sono istanze della classe JLabel
- Per creare un etichetta si scrive:

JLabel myLabel = new JLabel(testoLabel);

- dove testoLabel è una stringa che contiene il testo visualizzato dall'etichetta.

Etichette (2)

- I metodi utili della classe JLabel sono:

public String getText()

- restituisce il testo dell'etichetta

public String setText(String testo)

- Imposta il testo dell'etichetta, come specificato

Caselle di testo (1)

- Le caselle di testo contengono una riga di testo che può essere modificata sia dall'utente, sia dall'applicazione e solitamente sono utilizzate per le operazioni di input
- Le caselle di testo sono istanze della classe `TextField`, con costruttori:
 - `public TextField(String testo)` inizializza il testo
 - `public TextField(int colonne)` indica il numero di colonne
 - `public TextField(String testo, int colonne)` entrambi i precedenti

Caselle di testo (2)

- Metodi ereditati dalla classe JTextComponent:
 - *public void setText(String testo)* imposta il testo
 - *public String getTesto()* recupera il testo
 - *public String getSelectedText()* recupera il testo selezionato
 - *public void setEditable(boolean b)* rende la casella editabile o no

Pulsanti (1)

- I pulsanti, che solitamente contengono un etichetta sono usati per invocare un'azione quando sono premuti dall'utente, mediante il mouse.
- I pulsanti sono istanze della classe JButton
- Per creare un pulsante si scrive:
JButton myButton = new JButton(etichetta);
 - dove etichetta è una stringa che costituisce l'etichetta del pulsante

Pulsanti (2)

- Alcuni metodi utili ereditati dalla classe `AbstractButton`:
 - *`public void setText(String testo)`* imposta il testo
 - *`public String getText()`* recupera il testo
 - *`public void setEnabled(boolean b)`* abilita o disabilita, il pulsante, in base al valore del parametro.

Aggiungere un componente ad un contenitore (1)

- Si utilizza il metodo `add()` del Container, visto in precedenza.

```
myPanel.add(new JLabel("mia"));
```

```
myPanel.add(new TextField("testoIniziale"));
```

```
myPanel.add(new JButton("invia"));
```

Il modo con cui i componenti vengono sistemati è determinato dal layout manager.

Aggiungere un componente ad un contenitore (2)

- Una volta che un componente è stato creato, esso può essere aggiunto, attraverso la chiamata del metodo `add()` sul contenitore:

```
Container myContentPane= getContentPane();
```

```
myContentPane.add(new JButton("cancel"));
```

```
myContentPane.add(new JButton("go"));
```

Il modo con cui i componenti vengono sistemati è determinato dal layout manager.

Aggiungere un componente ad un contenitore (3)

```
private void initComponents() {  
    // TODO Auto-generated method st  
    this.setLayout(new FlowLayout());  
    lCognome = new JLabel("cognome:  
    this.add(lCognome);  
    tCognome = new JTextField(20);  
    this.add(tCognome);  
    lNome = new JLabel("nome: ");  
    this.add(lNome);  
    tNome= new JTextField(20);  
    this.add(tNome);  
    bInvia= new JButton("Invia");  
    this.add(bInvia);  
    lCompleto = new JLabel("Cognome-
```

Layout