

Progettazione, Algoritmi e Computabilità

Java Swing

Michele Beretta

michele.beretta@unibg.it



Java Swing

Java Swing

Swing è l'API (Application Programming Interface) Java che permette la creazione di interfacce visuali. Si basa sul modello MVC.

Caratteristiche principali:

- Altamente personalizzabile ed estendibile.
- Multi-piattaforma.
- Bassi requisiti computazionali.

Alternative a Swing

Esistono diverse alternative a Swing. Le principali sono:

- JFC/AWT (Abstract Window Toolkit)
 - La accenneremo ma non la useremo
 - Implementazione «vecchia», risalente a Java 1.0
 - **Swing non è un completo sostituto di AWT**: alcune funzionalità delle GUI sono ancora gestite tramite AWT (eventi, layout, etc.)
- SWT (Standard Widget Toolkit)
 - API grafica sviluppata dal team di Eclipse
 - Da molti ritenuta superiore (per funzionalità) a Swing.
 - Non è ancora uno standard vero e proprio
 - Non è implementato su tutte le piattaforme

Gerarchia delle librerie Swing

La libreria Swing è contenuta nel package `javax.swing`. I nomi delle classi Swing iniziano tutti con una J, questo li distingue dalle controparti AWT.

Ad esempio, `JFrame` VS `Frame`.

Frame

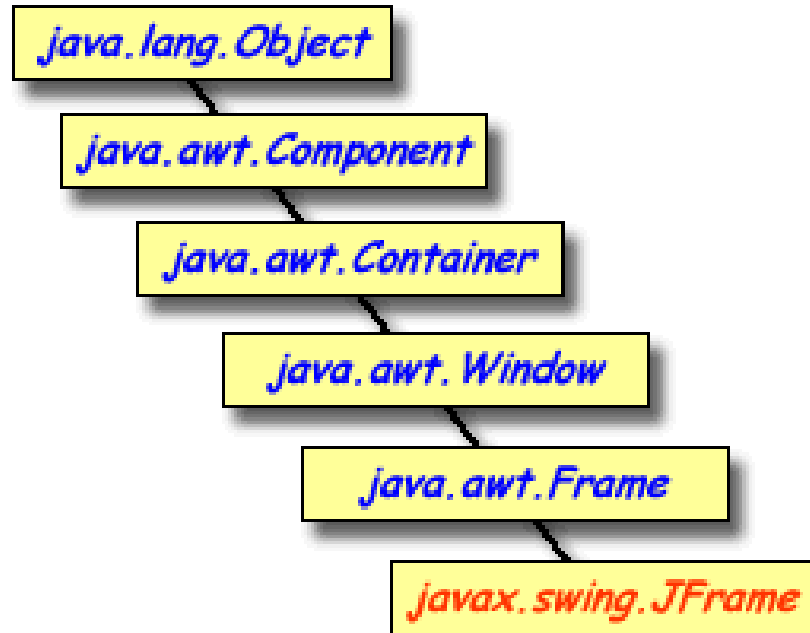
I **frame** (cornice, intelaiatura, scheletro) sono la “struttura” che contiene un’applicazione grafica.

Un frame fornisce un rettangolo “decorato” il cui compito consiste nel contenere tutto quello che l’applicazione deve mostrare all’utente.

I frame offrono in automatico una serie di funzionalità base:

- Pulsanti di riduzione a icona, massimizzazione e chiusura
- Riposizionamento e ridimensionamento tramite mouse
- Barra del titolo
- Icona dell’applicazione

Classi parent di JFrame



awt.Window

È un Container che può apparire sullo schermo come entità propria ma senza bordi, barre o controlli.

Possiede metodi per gestirne la posizione e la visibilità:

- `void setVisible(boolean b)`: mostra/nasconde la finestra (ereditato da `Component`).
- `void setLocation(int x, int y)`: posiziona la finestra alle coordinate indicate.
- `void setSize(int w, int h)`: dimensiona la finestra.
- `void toFront()`, `void toBack()`: aggiusta l'ordine di comparsa relativamente ad altre finestre.

awt.Frame

Un `Frame` è una `Window` a cui sono stati aggiunti bordi, titolo, pulsanti di controllo.

Oltre ai metodi di `Window`, un `frame` possiede metodi per gestire queste aggiunte:

- `void setTitle(String str)`: imposta un titolo.
- `void setResizable(boolean b)`: imposta la "ridimensionabilità" di una finestra.
- `void setIconImage(Icon icon)`: imposta l'icona della barra del titolo.

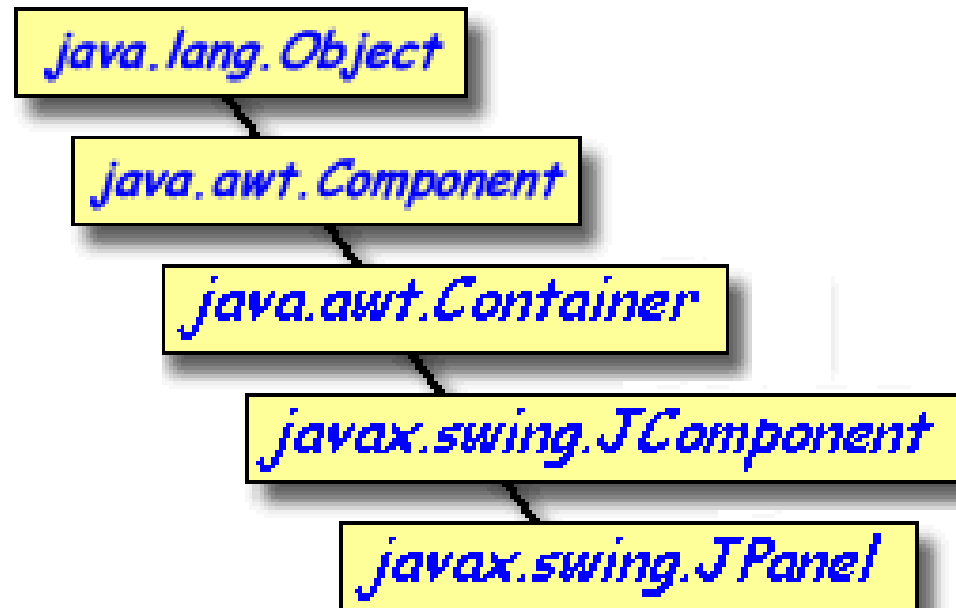
`javax.swing.JFrame`

Si tratta dell'effettivo frame che verrà utilizzato nelle applicazioni grafiche che tratteremo.

Esso è un Frame AWT a cui la libreria Swing aggiunge una serie di metodi importanti (anche per correggere malfunzionamenti di AWT). Un esempio di funzionalità aggiuntiva è quella dello `status`, ovvero la funzionalità che permette di ridurre ad icona o massimizzare l'applicazione:

- `void setExtendedState(int state)`: riduce a icona, massimizza e visualizza in modalità normale.
- `void setState(int state)`: analogo, ma non massimizza.

Classi parent di JPanel



`javax.swing.JComponent`

La classe `JComponent` introduce molte delle funzionalità legate a Swing.

La più importante è la possibilità, per un elemento, di essere disegnato attraverso il metodo `void paintComponent(Graphics g)`, che disegna dentro un `JComponent`.

È la classe base per quasi tutti i componenti Swing.

Per poter usare un componente che eredita da `JComponent`, occorre posizionare il componente in un container la cui radice sia un Container Swing (`JFrame`, `JDialog`, `JApplet`).

`javax.swing.JPanel`

È una delle più semplici componenti Swing. Serve come area rettangolare sulla quale:

- Inserire altri pannelli Swing
- Disegnare

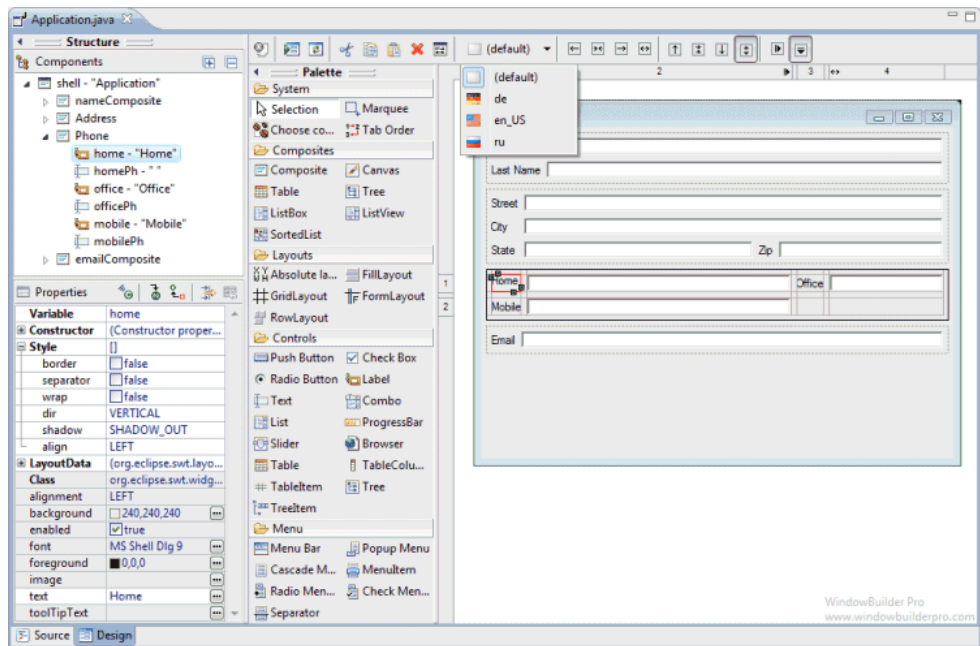
`javax.swing.JOptionPane`

La classe `JOptionPane` è la classe usata per implementare finestre di dialog.

```
JOptionPane.showMessageDialog(null,  
    "Message", "Title", JOptionPane.ERROR_MESSAGE);  
  
JOptionPane.showConfirmDialog(null,  
    "Message", "Title", JOptionPane.YES_NO_OPTION);  
  
String result = JOptionPane.showInputDialog("Prompt");
```

Eclipse Window Builder

WindowBuilder è un designer per SWT e Swing, integrato in Eclipse, che facilita la creazione di GUI Java, tramite l'utilizzo del drag-and-drop di componenti all'interno di un frame.



Java2D e Graphics

La classe `Graphics` e il metodo `paintComponent()` consentono di disegnare e/o scrivere.

Il metodo `paintComponent()` è ereditato da `JComponent`. Affinché ogni componente faccia ciò che si desidera, il metodo va ridefinito.

```
public void paintComponent(Graphics g) {  
    // Per ereditare le operazioni standard  
    super.paintComponent(g);  
  
    // Seguono istruzioni specifiche di disegno  
}
```

Graphics

Graphics è una classe astratta. Essa viene specializzata da:

- i vari OS in funzione delle loro caratteristiche di disegno
- i vari Component che hanno diritto ad un contesto grafico a seconda delle necessità e dei limiti di ciascuno

Si possono così programmare operazioni grafiche su tutte i componenti.

Graphics

- **Chi chiama paintComponent?**

Viene chiamato automaticamente ogni qualvolta sia necessario. Non deve essere chiamato manualmente!

- **Come forzare la chiamata del metodo paintComponent?**

`void repaint()`: forza il ridisegno della finestra

- **Quali azioni attivano paintComponent?**

Il metodo viene attivato da tutte le operazioni che provocano un ridisegno della finestra, come ridimensionamento della finestra, riduzione ad icona, massimizzazione, sovrapposizione di finestre, etc.

Graphics

Per lavorare `paintComponent` ha bisogno di un oggetto `Graphics` (contesto grafico). Esso è la sua «memoria» e il suo «libretto di istruzioni» per eseguire i comandi di disegno.

Un oggetto `Graphics` viene creato da `JComponent` appena deve essere disegnato la prima volta e mantiene le seguenti informazioni:

- l'oggetto sul quale si disegna
- il sistema di coordinate adottato per disegnare
- il colore di foreground (colore con cui disegnare)
- il font usato per le stringhe e le sue proprietà
- il «clip» (ritaglio)
- la modalità di disegno (Paint o XOR)
- la modalità di trasparenza e di blending

- etc.

Sistema di riferimento gli oggetti Graphics



- Si parte da in alto a sinistra
- La y va verso il basso
- Le misure sono in pixel

Classi per oggetti Graphics

Alcune classi importanti quando si lavora con oggetti Graphics sono:

- `Color` (definizione con RGB, costanti della classe): può essere utilizzata per impostare il colore di sfondo, il colore in primo piano o un colore generale.
- `Font` (definizione della dimensione del carattere, font-family, stile del testo): può essere utilizzata per definire il font del contesto grafico.
- Disegni vari (forme elementari, linee, spezzate, poligoni).
- `Toolkit` e `Image`: per la gestione di immagini.