

Graphics

Un oggetto di tipo Graphics rappresenta essenzialmente una porzione di schermo in cui è possibile mostrare testi, immagini, forme, in modo indipendente dalla piattaforma

Per disegnare su un oggetto GUI (di tipo *contenitore*), dobbiamo chiamare opportuni metodi sull'oggetto Graphics dello stesso.

Esistono due modi:

- ottenere l'oggetto Graphics con `oggettoGUI.getGraphics()`
- modificare il metodo di disegno del contenitore cioè **estendere la classe** ereditando da una classe Frame o Container o Panel (package **awt**) o JPanel (**swing**)

Non solo l'applet ha il metodo paint che può essere ridefinito, ma anche gli oggetti di tipo Component, ovvero tutti i componenti GUI (del package **awt**)¹, quindi per cambiare l'aspetto grafico di un qualsiasi componente grafico basta **ridefinirne**² il metodo paint.

Tra tutti i Component c'è la *tela* su cui è possibile disegnare, essa è un oggetto della classe Canvas.

Chi è abituato a programmare con altri linguaggi si aspetterà che nel metodo paint ci sia un' inizializzazione del *device* su cui si va a disegnare e poi una serie di istruzioni tipo drawLine, drawBox eccetera [1], ma questo non è del tutto vero, ovvero nei Component e nelle applet non c'è nessun metodo grafico.

Il parametro g di tipo Graphics del metodo paint o paintComponent, infatti, contiene i metodi grafici usabili da Java e rappresenta in qualche modo l'area (**contesto grafico**) in cui verranno disegnate le varie primitive.

I metodi paint (Graphics g) e paintComponent(Graphics g) *non possono essere invocati direttamente*.

Per disegnare un oggetto di tipo Graphics utilizzando ad esempio il metodo paint() della classe Component, passando come parametro il contesto grafico di tipo Graphics, si crea una classe che estende il Frame dentro il quale si vuole disegnare:

```
import java.awt.Frame;
import java.awt.Graphics;

public class SimpleGrafica extends Frame {
    public void paint (Graphics g) {
        g.drawRect(10, 10, 100, 75); // x, y, larghezza, altezza
    }
}
```

Graphics è una classe che *non può essere istanziata*: nel metodo paint o paintComponent viene ricevuto un oggetto di tale classe come parametro e se ne possono usare i metodi oppure si può recuperare il **contesto grafico** (di un contenitore: di tipo Frame o Container o Panel in awt o **JPanel** in swing) che possiede tutti i metodi di disegno.

Per ottenere il contesto grafico ad esempio della finestra corrente usando il metodo getGraphics():

```
Graphics g = getGraphics();
```

nb: usando il metodo getGraphics() sarà necessario richiamare il metodo dispose() per deallocare l'oggetto quando se ne termina l'uso.

[1] nel disegnare in ambiente Dev-Cpp:

```
int GraphDriver = DETECT, // rileva la più alta risoluzione possibile tipicamente
// VGA (640 x 480) ed equivale a GraphDriver=0
GraphMode; // auto-detected cioè GraphMode=0 tipicamente VGAHI

int Xpixel= 640, Ypixel=480;
initgraph (&GraphDriver, &GraphMode, "", Xpixel, Ypixel); // Start Window con inizializzazione device
setcolor(CYAN); // funzione per settare il colore impostato per gli assi
```

¹ Analogamente per gli oggetti di tipo JComponent (**swing**) esiste il metodo paintComponent

² <http://docs.oracle.com/javase/tutorial/2d/>