

## La componente per gestire un'area di disegno

La classe **Canvas**, nel package **java.awt**, rappresenta un'area rettangolare dello schermo su cui è possibile tracciare linee, disegnare rettangoli e cerchi di vari colori, visualizzare stringhe e immagini.

La classe *Canvas* contiene il metodo **paint** che viene richiamato automaticamente quando l'area di disegno deve essere visualizzata. Questo metodo contiene tutte le istruzioni che specificano che cosa deve essere rappresentato nell'area rettangolare.

La classe *Canvas* non viene utilizzata direttamente, ma viene invece creata una sottoclasse e, in questa, viene sovrascritto il metodo *paint*, per adattarlo alle esigenze dell'applicazione. Questa sottoclasse rappresenta una **componente personalizzata** e, in quanto sottoclasse della classe *Component*, può essere aggiunta a un contenitore per essere visualizzata.

La struttura generale di una sottoclasse di *Canvas* è la seguente:

```
import java.awt.*;

class Area extends Canvas
{
    public void paint(Graphics g)
    {
        // metodi per disegnare nell'area
    }
}
```

L'istruzione di *import* è necessaria perché le due classi, *Canvas* e *Graphics*, sono contenute nel package *java.awt*.

Il metodo *paint* ha come parametro un oggetto di classe **Graphics**. Quest'oggetto rappresenta concretamente l'area di disegno e offre tutti i metodi per disegnare all'interno della *Canvas*.

La classe *Graphics* gestisce un insieme di informazioni sullo stato dell'area di disegno, necessarie per le operazioni di disegno. Tra queste ci sono le informazioni sul colore corrente e sul font corrente, usati per disegnare le figure e per visualizzare le stringhe, che possono essere modificati con i seguenti metodi:

- **setColor(Color)**: modifica il colore corrente, impostando il nuovo colore specificato dall'attributo,
- **setFont(Font)**: modifica il font corrente, impostando il nuovo font specificato dall'attributo.

Per esempio, per impostare il colore blu e un font grassetto di dimensione 18 si usano i seguenti comandi:

```
g.setColor(Color.blue);
g.setFont(new Font("Dialog", Font.PLAIN, 18));
```

Il metodo **drawLine(int, int, int, int)** disegna una linea usando il colore corrente. Riceve come parametri 4 numeri interi che rappresentano le coordinate del punto iniziale e finale della linea. Per esempio, per disegnare una linea dal punto p1=(10,5) al punto p2=(40,95) si invoca il metodo *drawLine* sull'oggetto *g* nel seguente modo:

```
g.drawLine(10,5,40,95);
```

Per aggiungere una parte di testo a un'area grafica, deve essere utilizzato il metodo **drawString(String, int, int)**. Il primo parametro rappresenta il testo da visualizzare, mentre gli altri due rappresentano le coordinate del punto dove viene visualizzato il testo.

## ESEMPIO

### Disegnare 5 linee parallele e scrivere un messaggio.

Il programma è composto da due classi:

- *Area*: è la sottoclasse della classe *Canvas*, in cui viene ridefinito il metodo *paint* per disegnare il contenuto dell'area di disegno;
- *Disegno*: contiene il metodo *main* e ha il compito di creare una finestra inserendo l'area di disegno.

Il codice completo dell'applicazione è riportato di seguito.

### IMPLEMENTAZIONE DELLA CLASSE (*Area.java*)

```
import java.awt.*;

class Area extends Canvas
{
    public void paint(Graphics g)
    {
        g.setColor(Color.red);

        for(int i=1; i<=5; i++)
        {
            g.drawLine(50, 10*i, 200, 20+10*i);
        }

        g.setColor(Color.blue);
        g.setFont(new Font("Dialog", Font.PLAIN, 18));
        g.drawString("LINEE PARALLELE", 50, 100);
    }
}
```

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

```
import java.awt.*;

class Disegno
{
    public static void main(String argv[])
    {
        Frame f = new Frame("Disegno");
        Area a = new Area();

        f.setSize(300,200);
        f.setLocation(100,100);
        f.add(a);
        f.setVisible(true);
    }
}
```

