

GUI in Java con l'AWT – 2: Canvas e paint [opzionale]

Stefano Mizzaro

Dipartimento di matematica e informatica
Università di Udine
<http://www.dimi.uniud.it/~mizzaro>
mizzaro@dimi.uniud.it
Programmazione, lezione 23
12 maggio 2004

Canvas

- "Tela" (del pittore)
- Generico componente per disegnare
- Qualsiasi aspetto
- Uso: da sovrascrivere
- Bisogna prima capire come vengono disegnati i componenti dell'AWT

Stefano Mizzaro - AWT2

2

Disegno dei componenti

- Gestito dall'AWT (no interruzioni)
 - `repaint`: "metti in coda", non "ridisegna ora"
- Come disegna un componente?
 - Prima cancella lo sfondo
 - Poi disegna (`paint`)
- Chi disegna i componenti del AWT (`Button`, `Label`, ...)?
 - Ogni componente ha il suo metodo `paint` completamente gestito dall'AWT

Stefano Mizzaro - AWT2

3

Disegno di (sottoclassi di) Canvas

- Decidiamo noi come la nostra sottoclasse di `Canvas` va disegnata e mettiamo il codice in `paint`
- Quindi, sovrascriviamo `paint` (che verrà invocato al bisogno dall'AWT)
- Se vogliamo forzare il disegno, non invocheremo `paint`, ma `repaint` (→ `update = "clear" + paint`)

Stefano Mizzaro - AWT2

4

paint

- Codice da eseguire quando si deve ridisegnare il componente
- Per i componenti dell'AWT è già implementato
- Per disegnare quello che vogliamo (`Canvas`), bisogna sovrascriverlo
- Non va mai invocato (`toString`)
- Ha un argomento di tipo `Graphics`...

Stefano Mizzaro - AWT2

5

Un'altra classe ausiliaria: Graphics

- `void paint(Graphics g)`
- Il "contesto grafico": dove il componente disegna
- Metodi di `Graphics` (da usare in `paint`):
 - `drawLine(int x, int y, int x1, int y2)`
 - `drawOval(int x, int y, int w, int h)`
 - `drawRect(int x, int y, int w, int h)`
 - `drawString(String str, int x, int y)`
 - `fillOval, fillRect, ...`

Stefano Mizzaro - AWT2

6

Esempio di uso di Canvas (1/2)

```
import java.awt.*;
public class ProvaPaint {
    public static void main (String[] args) {
        Frame f = new Frame("Prova Paint");
        f.add(new Button("Nord"), BorderLayout.NORTH);
        f.add(new Button("Est"), BorderLayout.EAST);
        f.add(new Button("Ovest"), BorderLayout.WEST);
        f.add(new Button("Sud"), BorderLayout.SOUTH);
        Canvas c = new MyCanvas();
        c.setBackground(Color.BLUE);
        f.add(c, BorderLayout.CENTER);
        f.pack();
        f.setVisible(true);
    }
}
```

Stefano Mizzaro - AWT2

7

Esempio di uso di Canvas (2/2)

```
class MyCanvas extends Canvas {
    public void paint(Graphics g) {
        System.out.println("Invocato il paint");
        for(int i = 0; i <= 25; i = i + 5)
            g.drawOval(i*2, i*2, getWidth()-i*4,
                      getHeight()-i*4);
        g.drawString("Pippo", getWidth()/2, getHeight()/2);
        g.fillRect(getWidth()-20, getHeight()-20, 20, 20);
    }
    public Dimension getMinimumSize() {
        return new Dimension(100,100);
    }
    public Dimension getPreferredSize() {
        return getMinimumSize();
    }
}
```

Stefano Mizzaro - AWT2

8

Canvas: metodi importanti

- Sovrascrive **Component**:
 - void paint(Graphics g)
- Ereditati da **Component**:
 - public Dimension getMinimumSize()
 - public Dimension getPreferredSize()

Stefano Mizzaro - AWT2

9

Canvas e immagini

- Su un **Canvas** si può mostrare un'immagine
 1. Ottenere un oggetto **Image** da un file
 2. Usare **drawImage(Image, ...)** di **Graphics** nel **paint** di **Canvas**

```
class MyCanvas extends Canvas {
    Image img;
    public MyCanvas() {
        img =
            (Toolkit.getDefaultToolkit()).getImage(
                "unilogo.gif");
    }
    public void paint(Graphics g) {
        g.drawImage(img, 0, 0, this);
    }
    // ...come prima
}
```

Stefano Mizzaro - AWT2

10