

APPLICAZIONI & APPLET

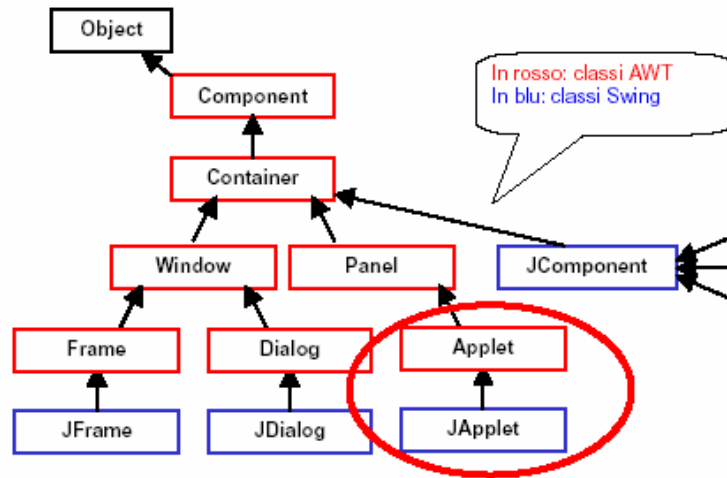
- Una *applet* ("applicazioncina") è una applicazione *non autonoma*, ma pensata per *far parte di una pagina Internet*
 - Porta dinamicità alle pagine HTML "statiche"
 - Viene *eseguita dal browser*, che quindi deve *incorporare un interprete Java*
 - Attenzione alla versione!
 - ·Molti browser supportano tuttora solo Java 1.1Per essere flessibili, il JDK installa il "*Java Plug in*", che consente al browser di *usare un interprete Java esterno (più aggiornato)*
-

APPLET

In quanto applicazione *non autonoma*, un'applet:

1. non deve creare un frame principale, perché usa la finestra del browser che la ospita (in effetti, Applet deriva direttamente da Panel e quindi è essa stessa un pannello)
 2. non ha un main, perché la sua vita è dipendente dalla pagina in cui è visualizzata
 3. è organizzata intorno a 4 metodi standard:
 1. `init()`, che gioca il ruolo del costruttore
 2. `start()` e `stop()`, chiamati dal browser ogni volta che occorre avviare /fermare l'applet
 3. `destroy()`, invocato quando il browser viene chiuso.
-

SWING: GERARCHIA DI CLASSI



Costruire un applet

- Per costruire un'Applet (con Swing), occorre:
 - creare una propria classe che estenda JApplet
 - implementare i metodi opportuni (tipicamente, almeno `init()`)
 - compilare l'applet nel modo standard
 - preparare una pagina HTML che la carichi
 - eseguire l'applet tramite l'*AppletViewer*
 - oppure –
 - aprire tale pagina con un browser

Esempio 1

```
import java.applet.*;
import java.awt.*;
import javax.swing.*;
public class Applet1 extends JApplet {
    Font f = new Font("Times", Font.BOLD, 36);
    public void paint(Graphics g) {
        g.setFont(f);
        g.setColor(Color.red);
        g.drawString("Ciao mondo!", 100, 50);
        // non implementa i metodi init, start, stop
    }
}
```

Esempio: la pagina HTML

```
<HTML><HEAD>
<TITLE> Applet Hello World </TITLE>
</HEAD> <BODY>
Questo e' cio' che produce la mia applet in
un rettangolo 500 x 100 (la scritta e' alle
coordinate 500,100 e si riferisce all'angolo
inferiore sinistro della stringa): <P>
<APPLET CODE="Applet1.class"
WIDTH=500 HEIGHT=100 >
<!--la classe dell'applet e le sue dimensioni-->
</APPLET>
</BODY>
</HTML>
```

Esempio 2

- Un'applet con tre pulsanti per cambiare il colore di sfondo
 - Il metodo `init()` fa le veci del costruttore: crea i componenti, installa i listener, etc.
- Non occorrono in questo caso gli altri metodi



Esempio 2

```
public class Applet2 extends JApplet {
    JButton redButton, blueButton, greenButton;
    JTextField messaggio;
    public void init() {
        Container c = getContentPane();
        // il contenitore che contiene la applet
        c.setBackground(SystemColor.window);
        c.setLayout(new FlowLayout());
        redButton = new JButton("Rosso");
        blueButton = new JButton("Azzurro");
        greenButton = new JButton("Verde");
        messaggio = new JTextField(26);
        messaggio.setText("Premere un pulsante");
        messaggio.setEditable(false);
        // i componenti non vanno aggiunti alla applet
        // ma al contenitore che la contiene
        c.add(messaggio); c.add(redButton);
```


APPLET: I METODI STANDARD

Un'applet è organizzata intorno a 4 metodi:

- **init()**, che viene chiamato dal browser quando lancia l'applet per la prima volta e quindi fa le veci di un costruttore, legge i parametri, etc
 - **start()**, che viene chiamato dal browser ogni volta che l'applet deve essere riavviata (perché torna visibile nella pagina)
 - tipicamente riavvia un'animazione o un thread
 - non occorre implementarlo se non ci sono animazioni o thread da riattivare
 - **stop()**, che viene chiamato dal browser ogni volta che l'applet deve essere fermata (perché esce dall'area visibile della pagina)
 - tipicamente ferma un'animazione o un thread
 - non occorre implementarlo se non ci sono animazioni o thread da fermare
 - **destroy()**, che viene chiamato dal browser quando il browser stesso si chiude
 - utile *in casi particolari*, per liberare i contesti grafici (di norma non occorre implementarlo)
-

DA APPLICAZIONE A APPLET

Come convertire un'applicazione in un'applet?

- eliminare il main che crea il frame: non serve più, il frame è quello del browser
 - sostituire JFrame con JApplet e assicurarsi che la classe sia *pubblica*, altrimenti l'applet non potrà essere caricata
 - eliminare la chiamata a setSize(): ora le dimensioni del frame sono decise dalla pagina HTML tramite HEIGHT e WIDTH
 - eliminare la chiamata a addWindowListener(): un'applet non può essere chiusa, termina quando l'utente esce dal browser
 - eliminare la chiamata a setTitle(): un'applet non ha titolo, è la pagina HTML che lo definisce
 - sostituire il costruttore col metodo init():
 - in realtà, un'applet può avere un costruttore, ma solo init() può recuperare i parametri tramite getParameter().
-

APPLET e SICUREZZA

- Un'applet non può fare tutto quello che fa una applicazione.
- Poiché può essere scaricata dalla rete, *sarebbe troppo pericoloso* permettere a un'applet di fare qualunque cosa.
- Un'applet è *costretta* a rispettare un ben preciso *modello di sicurezza* ("*sandbox*")
 - è eseguita in una "scatola" da cui non può uscire
 - non può contaminare (o spiare) i dati del computer dell'utente
- Un'applet di norma *non può*:
 - accedere al file system locale (neppure per leggere un file)
 - eseguire un altro programma
 - ottenere informazioni sull'utente
 - connettersi via rete a un computer *diverso da quello da cui è stata scaricata*
 - caricare la libreria Java, chiamare System.exit()

Applet firmate

- Attraverso tecnologie di cifratura, un'applet può essere *firmata*, ossia a essa può essere allegato un *certificato* che ne garantisce l'origine.
- Alle applet firmate, cui si attribuisce maggiore fiducia, *l'utente può consentire di svolgere alcune o tutte le operazioni sottoposte a vincolo*.
- Ogni browser può essere configurato per gestire le applet firmate.