

APPENDICE: APPLET

Java si è diffuso storicamente, e trae forza, dal concetto di *applet*¹ come piccola (almeno all'inizio dei tempi) applicazione da eseguirsi dentro un browser Internet

- *grafica portabile ed eseguibile ovunque*
- *modello di sicurezza "sandbox"*

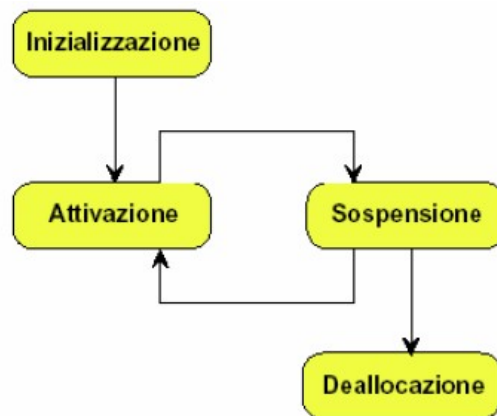
Una *applet* ("applicazioncina") è una applicazione *non autonoma*, ma pensata per *far parte di una pagina Internet*

- porta dinamicità alle pagine HTML "statiche"
- viene *eseguita dal browser*, che quindi deve *incorporare un interprete Java* e questo può causare un tempo di visualizzazione più alto

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

- **non ha un main**, perché la sua vita è dipendente dalla pagina in cui è visualizzata, **eredita** dalla [superclasse](#) ed è **public**.
- 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)
- è organizzata intorno a 4 [metodi standard](#):
 - **init()**, che gioca il ruolo del costruttore
 - **start()** e **stop()**, chiamati dal browser ogni volta che occorre avviare /fermare l'applet
 - **destroy()**, invocato quando il browser viene chiuso.

Diagramma degli [stati](#) di un'applet



Esistono *due versioni* di Applet:

- la classe Applet dell'AWT standard (*da Java 1.0 in poi*)
- la classe JApplet di Swing (*da Java 1.1.6 in poi*)

Se si usano componenti Swing, occorre necessariamente usare JApplet infatti una Applet con componenti Swing non viene disegnata correttamente

¹ Da <http://www33.brinkster.com/4binf/dispense/java/JavaApplet.pdf>

```
// AppletSaluto.java
```

```
import java.awt.Graphics; // importa solo la necessaria classe Graphics definita all'interno del package
                           // che implementa l'Interfaccia Utente detta Abstract Window Toolkit
```

```
public class AppletSaluto extends java.applet.Applet { // AppletSaluto è sottoclasse di Applet

    public void paint(Graphics g) {
        g.drawString("Ciao alla citta'!", 5, 25); // coordinate dell'angolo in basso
    } // a sinistra del testo
}
```

Con *file HTML* che fa riferimento al file bytecode dell'applet:

```
<object code="AppletSaluto.class" width="195"height="38"> </object>
```

Si ottiene il seguente effetto con uso di BROWSER (Internet Explorer recenti o FireFox Mozilla)

(5, 25) → **Ciao alla citta'!** *cioè una scritta nella finestra del browser*

```
// SecondoApplet.java con passaggio di parametri
```

```
import java.awt.*;
import java.applet.*;

public class SecondoApplet extends Applet {

    public void paint(Graphics g) {
        String nome = getParameter ("Nome");
        g.drawString("Ciao " + nome, 0, 50);
    }
}
```

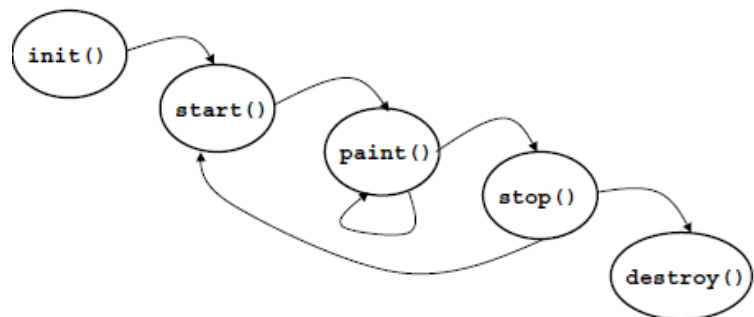
```
Con file HTML <html>
    <head><title> Saluto personalizzato</title></head>
    <body>
        <object code="SecondoApplet.class" width="195"height="38">
            <param name="Nome" value="Nome_proprio">
        </object>
    </body>
</html>
```

Si ottiene il seguente effetto con uso di BROWSER (Internet Explorer):

Ciao *Nome_proprio*

cioè un saluto personalizzato nella finestra del browser

La vita di un'Applet



Tutti i metodi sono vuoti tranne `paint` che viene ereditato da `Component`

// esempio di applet con descrizione degli Stati (*ridefinizione* di metodi originali)
 // da lanciare con **appletviewer** da linea di comando > appletviewer Stati.html oppure da **browser**²
 // Notare l'assenza del metodo main() infatti definiamo Stati come sottoclasse di Applet che viene
 // usata dalla classe principale predefinita AppletViewer al momento dell'esecuzione
 // oppure dalla **JVM integrata nel browser**

```
import java.applet.*;
```

```
public class Stati extends Applet {           // Stati è sottoclasse di Applet con costruttore Applet()

    public Stati ()    {                       // costruttore

        System.out.println("Invocato il costruttore di stati");
    }

    public void init ()    {                   // richiamato unica volta: quando carica l'Applet
                                                // cioè quando il browser lo preleva dal Web Server
        super.init(); // metodo originale
        System.out.println("Eseguito init()");
    }

    public void start () {                     // richiamato ogni volta che si accede a pagina web
                                                // invoca metodo paint(..)
        super.start(); // metodo originale
        System.out.println("Eseguito start()");
    }

    public void stop () {                      // richiamato ogni volta che si esce da pagina web
                                                // sia finestra iconizzata, sia non in primo piano
        super.stop(); // metodo originale
        System.out.println("Eseguito stop()");
    }

    public void destroy () {                  // richiamato quando si esce dall'appletviewer o da browser

        super.destroy(); // metodo originale
        System.out.println("Eseguito destroy()");
    }
}
```

Codice HTML salvato nel file **Stati .html**:

```
<html>
  <head><title>Stati di un Applet </title></head>
  <body>Lancia con appletviewer da linea di comando <P> &gt appletviewer Stati.html
  <P><object code ="Stati.class" width = 10 height = 10 ></object>
  <br>oppure seleziona da barra delle applicazioni: <em> Apri console</em>
  </body>
</html>
```

² Lanciando da browser aprire visualizzazione *console* (simbolo Plug-in Java che compare nella barra di sistema)
 NB: da pannello di controllo (Java Plug-in) si deve scegliere l'opzione "Mostra Java nella barra di sistema" e
 selezionare i browser in cui usare Java™ Plug-in come runtime Java predefinito