

# Corso sul linguaggio Java

## Modulo JAVA5

### A1.1 – **Awt** e contenitori

M. Malatesta A1.1-Awt e contenitori-05

1  
27/07/2011

## Prerequisiti

- Programmazione base in Java
- Utilizzo di classi e oggetti
- Elementi di un'interfaccia grafica

M. Malatesta A1.1-Awt e contenitori-05

2  
27/07/2011

# Introduzione

Iniziamo lo studio delle interfacce grafiche con la presentazione del *package* **AWT** e dei due contenitori grafici principali: finestre e pannelli.

Questi elementi sono oggetti software dei quali descriviamo attributi e principali metodi di utilizzo.

Nella prossima Unità A1.2 esaminiamo i principali controlli e nella successiva A1.3 vedremo come questi possano essere disposti nell'interfaccia utente

M. Malatesta A1.1-Awt e contenitori-05

3  
27/07/2011

## Il package **awt**

Per poter creare **un'interfaccia grafica** o **GUI** (*Graphical User Interface*) occorre importare il *package* grafico

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

Questo package:

- gestisce gli elementi grafici
- contiene le classi dei principali elementi grafici
- garantisce l'indipendenza dalla piattaforma

Pertanto saremo in grado di gestire le interfacce grafiche come **oggetti GUI**.

M. Malatesta A1.1-Awt e contenitori-05

4  
27/07/2011

# Caratteristiche di **awt**

Il *package* **awt** (*Abstract Windowing Toolkit*) è stato il primo tentativo di Java per rendere la grafica indipendente dalla piattaforma e dalle librerie del S.O. sottostanti.

Il *package* **awt** “incapsula” le **API** (*Application Program Interface*) del sistema operativo, per cui, per ogni elemento grafico tipico (ad es: pulsante) c'è una classe **awt** corrispondente (es: **java.awt.Button**).

Le operazioni di disegno del *bottone* sono perciò rimandate al corrispondente elemento *bottone* della libreria nativa detto *peer* (es: bottone Windows o bottone XLib)

M. Malatesta A1.1-Awt e contenitori-05

5  
27/07/2011

# Vantaggi e svantaggi di **awt**

## **Vantaggi:**

molto veloce perchè sfrutta su ogni sistema la libreria nativa (UNIX, Linux, Windows)

## **Svantaggi:**

le librerie di sistema sono molto diverse da una piattaforma all'altra:

- gli effetti grafici possono non essere gli stessi
- la resa grafica spesso è inaffidabile.

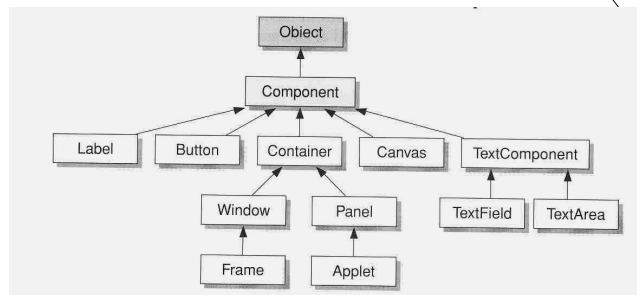
Bisognerà attendere la fine degli anni Novanta per la comparsa del *package* **Swing**.

M. Malatesta A1.1-Awt e contenitori-05

6  
27/07/2011

# La classe Component

Il package **awt** contiene le classi per la gestione degli elementi grafici; la gerarchia delle classi è mostrata in figura (parziale). La classe **Object** è la più generale e da essa ha origine la classe **Component**. Tra i componenti abbiamo la classe **Container**.



M. Malatesta A1.1-Awt e contenitori-05

7  
27/07/2011

# Contenitori e componenti

Gli elementi di una **GUI** si dicono **componenti** e possono essere di due tipi::

- **contenitori**
- **componenti** propriamente detti

I **contenitori** sono oggetti destinati a contenere i componenti secondo una certa disposizione grafica. Un classico esempio di contenitore è una finestra grafica.

I **componenti** sono oggetti di tipo grafico con i quali l'utente interagisce (pulsanti, menu, caselle di testo)

In questa Unità affrontiamo lo studio dai contenitori.

M. Malatesta A1.1-Awt e contenitori-05

8  
27/07/2011

# Contenitori

Un contenitore è una parte di un'interfaccia grafica ed è composto da vari elementi grafici come pulsanti, caselle di testo, ecc.

I contenitori sono:

1. le **Finestre** (classe **Frame**);
2. i **Pannelli** (classe **Panel**);
3. i **Canvas** (li vedremo quando tratteremo la grafica).

M. Malatesta A1.1-Awt e contenitori-05

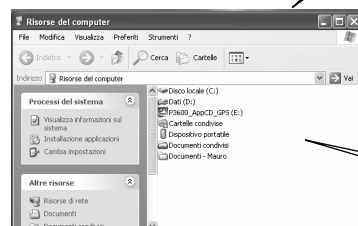
9  
27/07/2011

## 1. Finestre (**Frame**)

Gli elementi principali di una **finestra** sono:

- un'area in cui si disporranno i vari componenti
- una barra del titolo
- i 3 pulsanti di controllo nella barra del titolo

Vediamo alcuni dei metodi principali



Barra del titolo

Pulsanti di controllo

Area componenti

M. Malatesta A1.1-Awt e contenitori-05

10  
27/07/2011

# 1. Finestre (*Frame*)

```
import java.awt.*;
public class ProvaFrame
{ static public void main(String[] args)
{ Frame f1 = new Frame ("f1 (0,0,400,100)");
  Frame f2 = new Frame();
  f1.setSize (400, 100);
  f2.setBounds (100, 100, 400, 100);
  f2.setTitle ("f2 (100,100,400,100)");
  f2.setVisible (true);
  f2.setResizable (false);
}
}
```

Due tipi di costruttori

Imposta le dimensioni

Imposta posizione (100, 100) larghezza (400) e altezza (100)

Imposta il titolo

# 1. Finestre (*Frame*)

I principali metodi della classe **Frame** sono:

METODO	EFFETTO
<code>Frame ident = new Frame();</code>	Crea la finestra <i>ident</i>
<code>Frame ident = new Frame(String titolo);</code>	Crea la finestra <i>ident</i> con <i>titolo</i>
<code>f.setLocation (int riga, int colonna);</code>	Posiziona la finestra in ( <i>riga,colonna</i> )
<code>f.setSize (int larghezza,int altezza);</code>	Imposta <i>larghezza</i> ed <i>altezza</i> della finestra
<code>f.setVisible (boolean v)</code>	<i>v</i> = <b>false</b> (finestra invisibile), <i>v</i> = <b>true</b> (finestra visibile)
<code>f.setResizable (boolean r)</code>	<i>r</i> = <b>false</b> (ridimensionabile), <i>r</i> = <b>true</b> (non ridimension.)
<code>f.setBounds(int riga, int colonna, int larghezza,int altezza)</code>	Posiziona la finestra e imposta le dimensioni
<code>close ()</code>	Chiude la finestra

# 1. Finestre (*Frame*)

```
import java.awt.*;
public class Frame1 extends Frame
{
    public Frame1 ()
    {
        super ("Prima Finestra");
        setLocation (100,100);
        setSize (200,100);
        setVisible (true);
    }
    public static void main(String[] arg)
    {
        new Frame1();
        System.out.println("Ho creato la finestra");
    }
} // end class
```

M. Malatesta A1.1-Awt e contenitori-05

L'applicazione utilizza l'ereditarietà dalla classe **Frame**

I metodi ereditati possono essere invocati senza creare esplicitamente un oggetto di classe **Frame**

**Object**

**Frame1**

Gerarchia di contenimento

13  
27/07/2011

# 2. Pannelli (*Panel*)

```
import java.awt.*;
public class ProvaPanel
{
    public static void main(String args[])
    {
        Frame f = new Frame("Panel example");
        Panel p = new Panel();
        f.setSize(300, 200);
        p.setBackground(Color.green);
        f.add(p);
        f.setVisible(true);
    }
}
```

Il metodo **add** consente di inserire in un contenitore un elemento grafico (che a sua volta può essere un altro contenitore).

M. Malatesta A1.1-Awt e contenitori-05

**Object**

**ProvaPanel**

**Frame**

**Panel**

Gerarchia di contenimento

14  
27/07/2011

## 2. Pannelli (*Panel*)

I principali metodi per la classe **Panel** sono:

METODO	EFFETTO
<b>Panel</b> <i>ident</i> = new <b>Panel</b> ()	Crea il pannello <i>ident</i>
<b>setBackground</b> ( <b>Color</b> )	Imposta colore di sfondo

M. Malatesta A1.1-Awt e contenitori-05

15  
27/07/2011

## 2. Pannelli (*Panel*)

La classe **Color** contiene le seguenti costanti *statiche* per i colori

COSTANTE	COLORE
<b>Color.black</b>	Nero
<b>Color.blue</b>	Blu
<b>Color.cyan</b>	Cyano
<b>Color.darkGreen</b>	Verde scuro
<b>Color.gray</b>	Grigio
<b>Color.green</b>	Verde
<b>Color.lightGray</b>	Grigio chiaro
<b>Color.magenta</b>	Violetto
<b>Color.orange</b>	Arancio
<b>Color.pink</b>	Rosa
<b>Color.red</b>	Rosso
<b>Color.white</b>	Bianco
<b>Color.yellow</b>	Giallo

M. Malatesta A1.1-Awt e contenitori-05

16  
27/07/2011



# Programmazione ad eventi

È da notare che le interfacce grafiche create con gli esempi forniti e con la classe **awt** non sono interattive, ossia non consentono l'interazione con l'utente, ma si limitano a rappresentare a video i potenziali controlli utente.

Dopo illustrate le classi grafiche dei vari componenti, vedremo come, attraverso la **programmazione ad eventi**, sia possibile “animare” i controlli presenti su un'interfaccia grafica.

M. Malatesta A1.1-Awt e contenitori-05

17  
27/07/2011

## Argomenti

- Caratteristiche di **awt**
- Vantaggi e svantaggi di **awt**
- La classe **Component**
- Contenitori e componenti
- Contenitori
- 1. Finestre (**Frame**)
- 2. Pannelli (**Panel**)
- Programmazione ad eventi

M. Malatesta A1.1-Awt e contenitori-05

18  
27/07/2011

## Altre fonti di informazione

- P.Gallo, F.Salerno – Informatica Generale 1, ed. Minerva Italica
- M.Romagnoli, P.Ventura – Linguaggio C/C++, ed. Petrini
- M. Bigatti – Il linguaggio Java, ed. Hoepli