

Corso sul linguaggio Java

Modulo L6 (JAVA9)

4-Disegni Animazioni
Immagini Suoni

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

1
10/02/2012

Prerequisiti

- Funzioni grafiche in Java
- Utilizzo package awt

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

2
10/02/2012

Introduzione

In questa Unità utilizziamo Java per creare applet che sono in grado di riprodurre disegni, immagini e suoni.

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

3
10/02/2012

Disegni

ATTIVITA': scrivere il codice Java e il corrispondente HTML di un'applet che disegni un quadrato inscritto in un cerchio.

```
import java.awt.*;
import java.applet.*;
public class QuadratoInscritto extends Applet
{   int R, G, B;
    public void paint (Graphics g)
    {   imposta colore casuale e disegna rettangolo pieno;
        imposta colori casuali e disegna cerchio pieno;
    }
}
```

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

4
10/02/2012

Disegni

imposta colore casuale e disegna rettangolo pieno;
public void paint (Graphics g)
{ ColoriCasuali(); **g.setColor(new Color(R,G,B));**
 g.fillRect(20,20,60,60);
 ColoriCasuali(); **g.setColor(new Color(R,G,B));**
 g.fillOval(20,20,60,60);
}

imposta colori casuali e disegna cerchio pieno;
void ColoriCasuali()
{
 R=(int) Math.floor (Math.random() * 256);
 G=(int) Math.floor (Math.random() * 256);
 B=(int) Math.floor (Math.random() * 256);
}

Definisce in modo casuale il valore dei colori fondamentali

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

5
10/02/2012

Disegni

<HTML>
<HEAD>
 <TITLE>Quadrato inscritto in un cerchio</TITLE>
</HEAD>
<BODY>
 <APPLET code="QuadratoInscritto.class" width=250 height=200>
 </APPLET>
</BODY>
</HTML>

Si può facilmente verificare che, ogni volta che si iconizza o ridimensiona la finestra dell'applet, viene rieseguito il *repaint()* con colori diversi.

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

6
10/02/2012

Disegni

ATTIVITA': scrivere il codice Java e il corrispondente HTML di un'applet che simuli un proiettile con un cerchietto blu. Il click del mouse simula lo sparo e sposta il proiettile nel punto del click.

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class Pallino extends Applet
{   int X, Y;
    public void init( )
    {   gestoreMouse gm = new gestoreMouse( );
        addMouseListener(gm);
    }
```

Istanza il gestore del mouse e lo registra.

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

7
10/02/2012

Disegni

```
public void paint (Graphics g)
{   g.setColor (Color.blue);
    g.fillOval (X,Y,30,30);
}
private class gestoreMouse extends MouseAdapter
{   public void mousePressed (MouseEvent E)
    {   X = E.getX( );
        Y = E.getY( );
        repaint( );
    } // fine classe interna */
} /* fine classe pallino */
```

Disegna il pallino

Rileva la posizione attuale del mouse e ridisegna il pallino.

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

8
10/02/2012

Disegni

Codice HTML

```
<HTML>
<HEAD>
</HEAD>
<BODY>
  <APPLET CODE="pallino.class
    WIDTH=400 HEIGHT=500>
  </APPLET>
</BODY>
</HTM
```

Questo è il
semplice codice
HTML relativo

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

9
10/02/2012

Animazioni

ATTIVITA': scrivere il codice Java e il corrispondente HTML di un'applet che tracci una scritta animata che continuamente cambi colore.

```
importazione awt, awt.event, javax.swing.Timer, applet.*;
public class ScritteAnimate extends Applet
{ public void init() { creazione frame e oggetto Timer..... };
class oggettoTimer extends Canvas implements ActionListener
{ attributi: font, array di 255 colori, curcol, msec, timer;
  public oggettoTimer (int tempo) { costruttore } { ... }
  public void actionPerformed (ActionEvent e)
  { Quando scatta il timer, fa cambiare colore e ridisegna }
  public void paint (Graphics g) { imposta il font e disegna una parola }
}
```

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

10
10/02/2012

Animazioni

importazione awt, awt.event, javax.swing.Timer, applet.;*

```
import java.awt.*;  
import java.awt.event.*;  
import javax.swing.Timer;  
import java.applet.*;
```

Classe ScritteAnimate

```
public class ScritteAnimate extends Applet  
{ public void init()  
  { Frame f = new Frame("La girandola");  
    oggettoTimer t = new oggettoTimer(150);  
    t.setBackground(Color.white);  
    f.add(t);  
    f.setSize(new Dimension(300,200));  
    f.setVisible(true);  
  }  
}
```

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

11
10/02/2012

Animazioni

classe oggettoTimer: attributi

```
Font font = new Font("TimesRoman",Font.BOLD,24);  
Color color[] = new Color[255];  
int curcol = 0; // colore corrente  
int msec; // intervallo  
Timer timer; // crea effetto animazione  
  
public oggettoTimer (int tempo) //costruttore  
{ for ( int i = 0 ; i < color.length ; i++)  
  { color[i]= new Color((int)(Math.random()*1000)%256,  
    (int)(Math.random()*1000)%256, (int)(Math.random()*1000)%256);  
    msec=tempo; //tempo di permanenza scritta  
    timer = new Timer(msec,this);  
    timer.start();  
  }  
}
```

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

12
10/02/2012

Animazioni

```
public void actionPerformed (ActionEvent e) // cambio colore e ridisegna
{
    setForeground(color[curcol]); // quando scatta il timer...
    curcol++;                      //...incrementa il colore e...
    if (curcol == color.length )   //..se finiti colori...
        curcol=0;                 //...riparte da zero
}

public void paint (Graphics g) // imposta il font e disegna una parola
{
    g.setFont(font);
    g.drawString("Girandola "+ curcol,10,50);
}
```

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

13
10/02/2012

Immagini

ATTIVITA': scrivere il codice Java di un'applet che riproduca un'immagine data. Il codice corrispondente HTML è lasciato per esercizio.

```
import java.awt.*;           // per la classe Graphics
import java.applet.*;       // per la classe Applet
public class VisualizzaImmagine extends Applet
{
    Image ImmagineTorre;
    public void init( )
    {
        setBackground(Color.white);
        ImmagineTorre = getImage(getDocumentBase( ),"pisa.jpg");
    }
    public void paint (Graphics g)
    {
        g.drawImage(ImmagineTorre, 100, 100, this);
        getAppletContext( ).showStatus ("La torre di Pisa");
    }
} /* fine classe */
```

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

14
10/02/2012

Immagini

ATTIVITA': scrivere il codice Java di un'applet che riproduca un'immagine scegliendola casualmente tra un set di immagini date. Il codice corrispondente HTML è lasciato per esercizio.

```
import java.awt.*;           // per la classe Graphics
import java.applet.*;        // per la classe Applet
public class SfogliaImmagini extends Applet
{   Image Immagine[];        // array di immagini
    public void init()
    {   allocazione array e caricamento immagini;   }
    public void paint (Graphics g)
    {   dimensioni immagine e valore casuale;
        disegno immagine;           }
} /* fine classe */
```

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

15
10/02/2012

Immagini

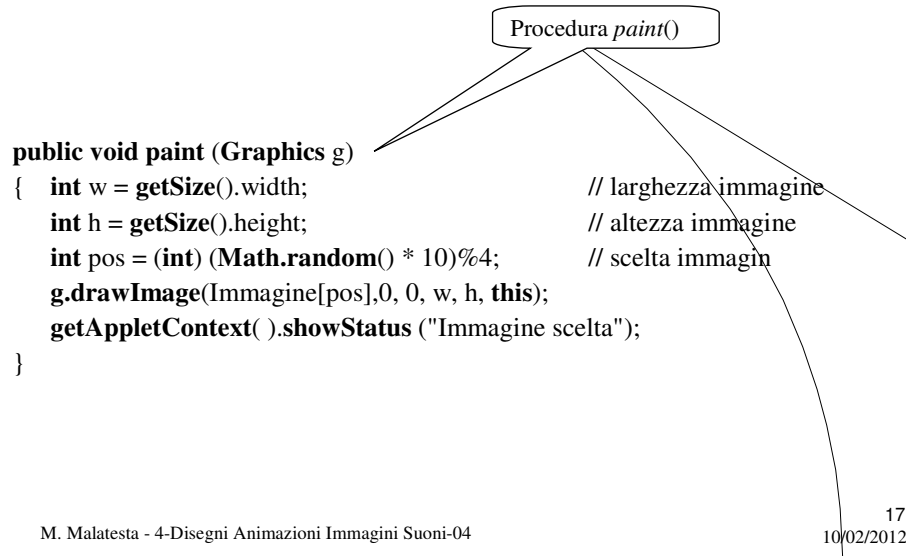
Procedura *init()*

```
public void init()
{   Immagine = new Image[4];    // crea e alloca array
    Immagine[0] = getImage(getDocumentBase(), "Acqua.jpg");
    Immagine[1] = getImage(getDocumentBase(), "Alberi.jpg");
    Immagine[2] = getImage(getDocumentBase(), "Oceano.jpg");
    Immagine[3] = getImage(getDocumentBase(), "Paradiso.jpg");
    setBackground(Color.white);
}
```

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

16
10/02/2012

Immagini



M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

Suoni

Per utilizzare le caratteristiche audio del calcolatore, Java offre nel package **applet** l'interfaccia **AudioClip** in grado di gestire:

- file sonori di tipo **AIFF, AU, WAV, MIDI, RMF, MPB**
- frequenze di campionamento da 8KHz a 48 KHz
- audio a 8 o 16 bit
- modalità mono o stereo

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

18
10/02/2012

Suoni

Un oggetto di classe **AudioClip** va creato mediante il metodo **getAudioClip()** dell'interfaccia **AppletContext**, vista in precedenza.

Esempio:

```
AudioClip ac = getAudioClip( getCodeBase(), nomefileaudio);
```

Sull'oggetto *ac* di classe **AudioClip** possiamo applicare il metodo **play()** come mostrato nel seguente esempio.

Indirizzo relativo del file audio. Per specificare un **indirizzo assoluto** new URL (http://indirizzo pagina)

Suoni

```
import java.applet.*;
public class SimpleAudioPlay extends Applet
{
    private AudioClip sound1, sound2;
    public void init()
    {
        sound1 = getAudioClip( getDocumentBase(), "uccelli.au" );
        sound2 = getAudioClip( getDocumentBase(), "pacman2.au" );
        sound1.loop();
        sound2.play();
    }
}
```

Dichiara due oggetti **AudioClip**

Inizializza i due oggetti **AudioClip**

Il package *applet*

Riassumiamo quanto visto finora riguardo al package **applet**.
Esso contiene le classi necessarie per creare applet e gestire il loro **contesto** (l'ambiente in cui essa esegue), che sono:

- **AppletContext**: interfaccia per comunicare con l'ambiente dell'applet (link, immagini, suoni)
- **AudioClip**: interfaccia per riproduzione di suoni
- **Applet**: sottoclasse di **Panel** per la creazione di applet.

L'interfaccia **AppletContext**

Riassumiamo i metodi dell'interfaccia **AppletContext**, in particolare:

Metodi	Effetto
AudioClip getAudioClip (URL <i>url</i> , String <i>nomefile</i>)	Crea un oggetto musicale (<i>clip</i>) avente indirizzo <i>url</i> e nome esterno <i>nomefile</i>
Image getImage (URL <i>url</i>)	Crea un oggetto immagine con indirizzo <i>url</i> , che può essere stampato da paint()
void showStatus (String <i>status</i>)	Visualizza <i>status</i> nella riga di stato

L'interfaccia **AudioClip**

L'interfaccia **AudioClip** contiene i seguenti metodi:

Metodi	Effetto
void loop ()	Avvia la riproduzione continua di suoni
void play ()	Avvia la riproduzione di suoni
void stop ()	Arresta la riproduzione di suoni

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

23
10/02/2012

Argomenti

- Disegni
- Animazioni
- Immagini
- Suoni
- Il package applet
- L'interfaccia **Appletcontext**
- L'interfaccia **AudioClip**

M. Malatesta - 4-Disegni Animazioni Immagini Suoni-04

24
10/02/2012

Altre fonti di informazione

- P.Gallo, F.Salerno – Java, la programmazione a oggetti, ed. Min. Italica
- A.Lorenzi, a.Rizzi, Il linguaggio Java – ed. ATLAS, p. 262
- P.Camagni, R.Nikolassy, Java, ed. Hoepli, p. 187