

Corso sul linguaggio Java

Modulo JAVA5

C2.1 – Animazioni

M. Malatesta C2.1-Animazioni-23

1
10/04/2013

Prerequisiti

- Programmazione base in Java
- Utilizzo di classi e oggetti
- Utilizzo di elementi grafici ed eventi
- Utilizzo classe **Canvas**

M. Malatesta C2.1-Animazioni-23

2
10/04/2013

Introduzione

Le funzionalità grafiche di Java consentono di rappresentare, ad esempio in un **Canvas**, una sequenza di immagini ciascuna delle quali differisce di poco dalla precedente, creando un effetto di movimento che si chiama **animazione**.

L'animazione, o qualunque altro effetto **temporizzato**, fa uso della classe **Timer**.

All'inizio dell'Unità, illustriamo la classe **Timer** e, successivamente, esaminiamo gli strumenti che occorrono per creare vari effetti visivi o acustici.

M. Malatesta C2.1-Animazioni-23

3
10/04/2013

La classe Timer

Per creare l'effetto di animazione è necessario disporre della classe **javax.swing.Timer** che consente di creare un oggetto che, mandando segnali cadenzati (evento **actionPerformed()**), permette di temporizzare le operazioni.

Funzioni classe Timer	
METODO	EFFETTO
Timer (<i>int delay, ActionListener listener</i>)	Crea oggetto Timer che avvisa i suoi ascoltatori <i>listeners</i> ogni <i>delay</i> millisecondi
void start ()	Avvia il Timer , che inizia a mandare eventi ai suoi ascoltatori
void stop ()	Arresta il Timer e smette di inviare eventi
void setDelay (<i>int delay</i>)	Imposta il <i>delay</i>
int getDelay ()	Restituisce il valore di <i>delay</i>

M. Malatesta C2.1-Animazioni-23

4
10/04/2013

Tecnica di utilizzo del timer

Un'applicazione temporizzata prevede almeno l'implementazione di due classi:

Classe *oggettoTimer* che:

1. contiene il costruttore di un oggetto di classe **Timer**
2. implementa la classe di ascolto **actionListener**, per gestire gli eventi prodotti dal timer;
3. crea un ascoltatore **actionPerformed()** che ad ogni segnale proveniente dal timer produce (ad esempio) un beep, oppure effetti grafici o visivi cadenzati.

M. Malatesta C2.1-Animazioni-23

5
10/04/2013

Tecnica di utilizzo del timer

Classe *prova*, che:

1. crea un *frame* o un *panel* e ne imposta le caratteristiche
2. crea un oggetto *t* di classe *oggettoTimer*
3. registra *t* nel frame

Ovviamente, è possibile affiancare un'ulteriore classe *Adattatore* che gestisce la chiusura della finestra.

M. Malatesta C2.1-Animazioni-23

6
10/04/2013

La classe *oggettoTimer*

class oggettoTimer **extends** Canvas **implements** ActionListener

```
{  dichiarazione attributi;
  int msec;      // tempo di delay
  Timer t;      // oggetto timer
  public oggettoTimer (int delay)
  {
    msec=delay;
    inizializzazione attributi;
    t = new Timer (msec, this);
    t.start();
  }
  public void actionPerformed (ActionEvent e)
  {
    gestione evento;
    repaint();
  }
  public void paint (Graphics g)
  {
    metodi grafici;
  }
}
```

Per gestire
l'evento lanciato
dal timer

Attributi specifici del problema

Creazione oggetto **Timer**
per gestire eventi
cadenzati

Eventuali operazioni
grafiche

M. Malatesta C2.1-Animazioni-23

7
10/04/2013

La classe *prova*

public class prova

```
{  public static void main(String args[])
  {
    Frame f = new Frame("Esempio di animazione");
    f.setLocation(200, 300);
    f.setSize(500, 300);
    oggettoTimer t = new oggettoTimer(50);
    f.add(t);
    f.addWindowListener(new Adattatore());
    f.setVisible(true);
  } // end main
} // end class
```

Crea oggetto *t* di
classe
oggettoTimer
impostando il
delay a 50
millisecondi

Registra l'ascoltatore
per la chiusura della
finestra

ATTIVITA': scrivere la classe *Adattatore* che
contenga la sola implementazione del metodo
windowClosing()

M. Malatesta C2.1-Animazioni-23

8
10/04/2013

La classe *prova*

```
class Adattatore extends WindowAdapter
```

```
{  
    public void windowClosing (WindowEvent e)  
    {  
        System.exit (0);  
    }  
}
```

Ereditando da questa classe, si implementano i soli metodi di interesse

Chiude la finestra e termina

M. Malatesta C2.1-Animazioni-23

9
10/04/2013

Possibili utilizzi del timer

Tramite lo schema indicato è possibile ottenere effetti di temporizzazione per:

- eseguire operazioni cadenzate;
- animare testo, creando testo scorrevole in orizzontale, verticale o altro;
- animare immagini, creando effetti di movimento, come avviene nei fotogrammi cinematografici.

M. Malatesta C2.1-Animazioni-23

10
10/04/2013

Esempi di utilizzo del timer

1. il beep a tempo

ATTIVITA': scrivere un'applicazione che simuli un metronomo, mediante un *beep* a tempo emesso dal sistema, ogni secondo mediante un oggetto **Timer**.

L'applicazione prevede:

- una classe *oggettoTimer* il cui costruttore crea un oggetto di classe **Timer**, che consente la temporizzazione; in questa classe si crea anche un ascoltatore che contiene le istruzioni da eseguire ad ogni segnale del timer;
- una classe *Prova* che crea un oggetto di classe *oggettoTimer* e lo registra in un frame apposito.

M. Malatesta C2.1-Animazioni-23

11
10/04/2013

Esempi di utilizzo del timer

1. il beep a tempo

ATTIVITA': scrivere la classe *oggettoTimer* che simuli un metronomo, mediante un *beep* a tempo emesso dal sistema, ogni secondo mediante un oggetto **Timer**.

class oggettoTimer **extends** Canvas **implements** ActionListener

```
{ Timer t;  
  int delay;                //milliseconds  
  public oggettoTimer (int delay)  
  { t= new Timer (delay, this);  
    t.start();              // avvia il timer  
  }  
  public void actionPerformed (ActionEvent evt)  
  { gestione evento }
```

}M. Malatesta C2.1-Animazioni-23

12
10/04/2013

Esempi di utilizzo del timer

1. il beep a tempo

ATTIVITA': L'emissione del *beep* può essere ottenuta nei due modi seguenti:

```
public void actionPerformed (ActionEvent evt)
{
    //Toolkit.getDefaultToolkit().beep();
    System.out.println ("\007");
}
```

M. Malatesta C2.1-Animazioni-23

13
10/04/2013

Esempi di utilizzo del timer

1. il beep a tempo

ATTIVITA': scrivere la classe *Prova* per testare la classe *oggettoTimer*.

```
public class Prova
{
    public static void main(String args[])
    {
        Frame f = new Frame("Esempio di timer");
        f.setLocation(200, 300);
        f.setSize(500, 300);
        oggettoTimer t = new oggettoTimer(1000);
        f.add(t);
        f.setVisible(true);
    }
}
```

Crea un oggetto di classe *oggettoTimer* con tempo di *delay* iniziale di 1000 msec.

M. Malatesta C2.1-Animazioni-23

14
10/04/2013

Esempi di utilizzo del timer

2. animare testo

ATTIVITA': scrivere un'applicazione che scriva una stringa in un canvas, variandone in modo cadenzato il colore.

L'applicazione prevede:

- una classe *oggettoTimer* il cui costruttore crea un oggetto di classe **Timer**, che consente la temporizzazione, un array di colori casuali; in questa classe si crea anche un ascoltatore che cambia il colore corrente e riscrive la stringa ad ogni segnale del timer;
- una classe *Prova* che crea un oggetto di classe *oggettoTimer* e lo registra in un frame apposito.

M. Malatesta C2.1-Animazioni-23

15
10/04/2013

Esempi di utilizzo del timer

2. animare testo

ATTIVITA': scrivere una classe *oggettoTimer* che scriva una stringa, variandone in modo cadenzato il colore.

```
class oggettoTimer extends Canvas implements ActionListener
{
    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)
    {
        for ( int i = 0 ; i < color.length ; i++) // inizializza array color[ ];
            msec = tempo;                       //tempo di permanenza scritta
        timer = new Timer (msec,this);    timer.start();
    }
    public void actionPerformed (ActionEvent e) { ....}
    public void paint (Graphics g) { ... }
}
```

M. Malatesta C2.1-Animazioni-23

16
10/04/2013

Esempi di utilizzo del timer

2. animare testo

ATTIVITA': implementare il metodo **actionPerformed()** che imposta il colore del testo e incrementa *curcol*.

```
public void actionPerformed (ActionEvent e)
{
    setForeground (color[curcol]);
    curcol++;
    if (curcol == color.length )
        curcol=0;
}
```

// quando scatta il timer
//... cambia colore e ridisegna
// quando scatta il timer...
//...incrementa il colore e...
//..se finiti colori...
//...riparte da zero

M. Malatesta C2.1-Animazioni-23

17
10/04/2013

Esempi di utilizzo del timer

2. animare testo

ATTIVITA': implementare il metodo **paint()** che imposta il *font* e disegna la stringa.

```
public void paint (Graphics g)
{
    g.setFont(font);
    g.drawString("Girandola "+curcol,10,50);
}
```

M. Malatesta C2.1-Animazioni-23

18
10/04/2013

Esempi di utilizzo del timer

3. animare immagini

ATTIVITA': scrivere un'applicazione che data una serie di immagini, le mostri in sequenza, con cadenza data dal timer.

L'applicazione prevede:

- una classe *oggettoTimer* il cui costruttore crea un oggetto di classe **Timer**, che consente la temporizzazione, una sequenza di N oggetti immagine (ad es. **jpg**) e un attributo intero *prec* posto a 0, che selezioni, ad ogni istante, l'immagine da visualizzare.
- una classe *Prova* che crea un oggetto di classe *oggettoTimer* e lo registra in un frame apposito

M. Malatesta C2.1-Animazioni-23

19
10/04/2013

Esempi di utilizzo del timer

3. animare immagini

ATTIVITA': scrivere la classe *oggettoTimer* con attributi **Timer t**, **int msec** e **int prec** e le 4 immagini di *Duke* presenti nei file: file0.gif, file1.gif, file2.gif, file3.gif

class oggettoTimer **extends** Canvas **implements** ActionListener

```
{ Timer t;
  int msec;
  int prec=0;
  Image img0 = Toolkit.getDefaultToolkit().getImage ("file0.gif");
  Image img1 = Toolkit.getDefaultToolkit().getImage ("file1.gif");
  Image img2 = Toolkit.getDefaultToolkit().getImage ("file2.gif");
  Image img3 = Toolkit.getDefaultToolkit().getImage ("file3.gif");
  public oggettoTimer (int delay)
  { msec = delay;
    t = new Timer (msec, this);    t.start();
  }
  public void actionPerformed (ActionEvent e) { ... }
  public void paint (Graphics g) { ... }
```

} M. Malatesta C2.1-Animazioni-23

20
10/04/2013

Esempi di utilizzo del timer

3. animare immagini

ATTIVITA': implementare il metodo **actionPerformed()** che incrementa *prec* e ridisegna con il metodo **repaint()**.

```
public void actionPerformed (ActionEvent e)
{
    if (prec<3) prec++;           // incrementa conteggio e...
    else prec=0;
    repaint();                   //...ridisegna
}
```

M. Malatesta C2.1-Animazioni-23

21
10/04/2013

Esempi di utilizzo del timer

3. animare immagini

ATTIVITA': implementare il metodo **paint()** che cambia l'immagine con la successiva in base al valore di *prec* tramite una **switch**.

```
public void paint (Graphics g)
{ // crea oggetti di classe Image
    switch (prec)
    {
        case 0: g.drawImage(img0,50,50, this); break;
        case 1: g.drawImage(img1,50,50, this); break;
        case 2: g.drawImage(img2,50,50, this); break;
        case 3: g.drawImage(img3,50,50, this); break;
    }
}
```

M. Malatesta C2.1-Animazioni-23

22
10/04/2013

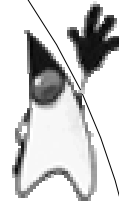
Esempi di utilizzo del timer

3. animare immagini

ATTIVITA': scrivere la classe principale *AnimazioneImmagini.java* che crei un frame con proprietà, imposti il colore di sfondo, crei un *oggettoTimer* e lo registri, che registri un adattatore per chiudere la finestra.

```
public class AnimazioneImmagini
{   public static void main(String args[] )
    {   Frame f = new Frame ("Animazione");
        f.setBackground(Color.lightGray);
        oggettoTimer t=new oggettoTimer (100);
        f.add(t);
        f.addWindowListener(new Adattatore());
        f.setLocation(200, 200);
        f.setSize(750, 400);
        f.setVisible(true);
    }
}
```

M. Malatesta C2.1-Animazioni-23



23
10/04/2013

Argomenti

- La classe **Timer**
- Tecnica di utilizzo del timer
- La classe *oggettoTimer*
- La classe *prova*
- Possibili utilizzi del timer
- Esempi di utilizzo del timer
 1. il beep a tempo
 2. animare testo
 3. animare immagini

M. Malatesta C2.1-Animazioni-23

24
10/04/2013