

Corso sul linguaggio Java

Modulo JAVA5

5.1.2 – Componenti

M. Malatesta 5.1.2 - Componenti-06

1
27/07/2011

Prerequisiti

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

M. Malatesta 5.1.2 - Componenti-06

2
27/07/2011

Introduzione

Ora che sappiamo creare i contenitori, vediamo quali oggetti grafici è possibile inserire in essi.

In questa Unità, descriviamo i più comuni oggetti grafici di Java che, essendo oggetti software, possiedono ciascuno proprietà e metodi caratteristici.

M. Malatesta 5.1.2 - Componenti-06

3
27/07/2011

Componenti

I **componenti** che esaminiamo sono i seguenti:

1. Etichette
2. Pulsanti
3. Caselle di testo
4. Aree di testo
5. Caselle di spunta
6. Caselle di opzione
7. Caselle combinate
8. Liste
9. Barre di scorrimento
10. Menu

M. Malatesta 5.1.2 - Componenti-06

4
27/07/2011

1. Etichette (Label)

```
import java.awt.*;
public class TestLabel
{
    public static void main(String args[])
    {
        Frame f = new Frame("Prova label");
        Label vuota = new Label();
        Label nome = new Label("Mario");
        Label completa = new Label ("Mario", Label.CENTER);
        Panel p = new Panel();
        f.setSize(300, 300);
        vuota.setBackground(Color.gray);
        vuota.setText("Testo");
        nome.setBackground(Color.red);
        nome.setText("Mario");
        completa.setForeground(Color.orange);
        completa.setBackground(Color.blue);
        p.setBackground(Color.lightGray);
        p.add(vuota); p.add(nome); p.add(completa); f.add(p);
        f.setVisible(true);
    }
}
```

Costruttori

Colore sfondo e testo

Inserisce pannello nel frame

Inserisce etichette nel pannello

5
27/07/2011

M. Malatesta 5.1.2 - Componenti-06

1. Etichette (Label)

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

METODO	EFFETTO
<code>Label ident = new Label();</code>	Crea l'etichetta <i>ident</i> vuota
<code>Label ident = new Label(String testo);</code>	Crea l'etichetta <i>ident</i> con <i>testo</i>
<code>Label ident = new Label(String testo, int all);</code>	Crea l'etichetta <i>ident</i> con <i>testo</i> allineato a destra (<i>all</i> =Label.RIGHT), centro (<i>all</i> =Label.CENTER) o sinistra (<i>all</i> =Label.LEFT) (oppure <i>all</i> =2,1,0)
<code>!setForeground (Color.colore);</code>	Imposta il <i>colore</i> del testo
<code>!setBackground (Color.colore);</code>	Imposta il <i>colore</i> dello sfondo
<code>!setText (String stringa);</code>	Imposta il testo nell'etichetta
<code>String !getText ();</code>	Restituisce il testo presente nella label
<code>int !getAlignment ();</code>	Restituisce l'allineamento corrente
<code>int !setAlignment (int all);</code>	Imposta allineamento dell'etichetta
<code>setFont (Font font);</code>	Imposta il <i>font</i> dei caratteri dell'etichetta

M. Malatesta 5.1.2 - Componenti-06

6
27/07/2011

2. Pulsanti (Button)

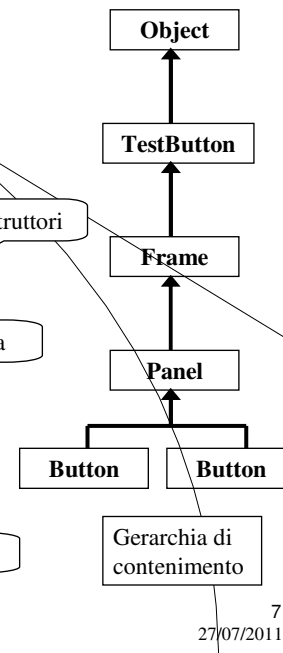
```
import java.awt.*;
public class TestButton
{
    public static void main(String args[])
    {
        Frame f = new Frame("Prova pulsanti");
        Button indietro = new Button("Indietro");
        Button avanti = new Button();
        Panel p = new Panel();
        f.setSize(300, 300);
        avanti.setLabel("Avanti");
        p.add(indietro);
        p.add(avanti);
        f.add(p);
        f.setVisible(true);
    }
}
```

Costruttori

Imposta etichetta

Inserisce pulsanti
in pannello

Inserisce pannello nel frame



M. Malatesta 5.1.2 - Componenti-06

7
27/07/2011

2. Pulsanti (Button)

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

METODO	EFFETTO
Button <i>ident</i> = new Button ();	Crea un pulsante <i>ident</i> senza etichetta
Button <i>ident</i> = new Button (String <i>testo</i>);	Crea un pulsante <i>ident</i> con etichetta <i>testo</i>
<i>b</i> .setLabel (String <i>testo</i>);	Imposta l'etichetta del pulsante <i>ident</i>
String <i>b</i> .getLabel ();	Restituisce l'etichetta presente nel pulsante
<i>b</i> .setEnabled (boolean <i>e</i>);	Abilita (<i>e</i> = true) o disabilita (<i>e</i> = false) il pulsante

M. Malatesta 5.1.2 - Componenti-06

8
27/07/2011

3. Caselle di testo (TextField)

```
import java.awt.*;
public class TestTextField
{ public static void main(String[] args)
  { Frame f = new Frame("Immissione dati");
    Panel p = new Panel();
    Label nome_lbl = new Label("Nome prodotto: ", Label.RIGHT);
    Label prezzo_lbl = new Label("Prezzo: ", Label.RIGHT);
    TextField nome_txt = new TextField("Immettere nome", 20);
    TextField prezzo_txt = new TextField("Immettere prezzo", 14);
    p.add(nome_lbl); p.add(nome_txt);
    p.add(prezzo_lbl); p.add(prezzo_txt);
    f.setLocation (100,100);
    f.setSize (300, 200);
    f.add(p); f.setVisible(true);
  }
} // end class
```

M. Malatesta 5.1.2 - Componenti-06

9
27/07/2011

3. Caselle di testo (TextField)

L'effetto dell'esempio precedente
è mostrato a fianco.



M. Malatesta 5.1.2 - Componenti-06

10
27/07/2011

3. Caselle di testo (TextField)

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

METODO	EFFETTO
TextField <i>ident</i> = new TextField ();	Crea una casella di testo <i>ident</i> vuota
TextField <i>ident</i> = new TextField (String <i>testo</i>);	Crea una casella <i>ident</i> con <i>testo</i>
TextField <i>ident</i> = new TextField (int <i>i</i>);	Crea una casella di testo <i>ident</i> vuota di lunghezza <i>i</i>
TextField <i>ident</i> = new TextField (String <i>testo</i> , int <i>i</i>);	Crea una casella di testo <i>ident</i> con <i>testo</i>
String <i>t</i> .getText ();	Restituisce il testo presente nella casella di testo <i>f</i>
<i>t</i> .setText (String <i>testo</i>);	Imposta il <i>testo</i> nella casella di testo <i>f</i>
<i>t</i> .setEditable (boolean <i>e</i>)	Imposta la casella <i>f</i> come editabile (<i>e</i> = true) o non editabile (<i>e</i> = false)

M. Malatesta 5.1.2 - Componenti-06

11
27/07/2011

4. Aree di testo (TextArea)

```
import java.awt.*;
public class TestTextArea
{
    public static void main(String[] arg)
    {
        Frame f = new Frame("Immissione dati");
        Panel p = new Panel();
        Label nome_lbl = new Label("Nome prodotto: ", Label.RIGHT);
        Label prezzo_lbl = new Label("Prezzo: ", Label.RIGHT);
        TextField nome_txt = new TextField("Immettere nome", 20);
        TextField prezzo_txt = new TextField("Immettere prezzo", 14);
        TextArea msg_txt = new TextArea("Commenti", 4,20);
        p.add(nome_lbl);      p.add(nome_txt);
        p.add(prezzo_lbl);    p.add(prezzo_txt);
        p.add(msg_txt);      f.setVisible (true);
        f.setLocation (100,100);
        f.setSize (300, 200);f.add(p); f.setVisible(true);
    }
} // end class
```

M. Malatesta 5.1.2 - Componenti-06

12
27/07/2011

4. Aree di testo (TextArea)

L'area di testo consente
l'immissione di testo su più linee.



M. Malatesta 5.1.2 - Componenti-06

13
27/07/2011

4. Aree di testo (TextArea)

I metodi più importanti della classe **TextArea** sono:

METODO	EFFETTO
<code>TextArea ident = new TextArea ();</code>	Crea un'area di testo <i>ident</i> vuota
<code>TextArea ident = new TextArea (String testo);</code>	Crea un'area di testo <i>ident</i> con <i>testo</i>
<code>TextArea ident = new TextArea (int r, int c);</code>	Crea un'area di testo <i>ident</i> vuota di <i>r</i> righe e <i>c</i> colonne
<code>TextArea ident = new TextArea (String testo, int r, int c);</code>	Crea un'area di testo <i>ident</i> con <i>testo</i> su <i>r</i> righe e <i>c</i> colonne
<code>String a.getText ();</code>	Restituisce il testo presente nell'area di testo <i>a</i>
<code>a.setText (String testo);</code>	Imposta il <i>testo</i> nell'area di testo <i>a</i>
<code>a.setEditable (boolean e)</code>	Imposta l'area di testo <i>a</i> come editabile (<i>e=true</i>) o non editabile (<i>e=false</i>)
<code>a.appendString (String testo)</code>	Aggiunge ad <i>a</i> il <i>testo</i>

M. Malatesta 5.1.2 - Componenti-06

14
27/07/2011

5. Caselle di spunta (Checkbox)

```
import java.awt.*;
public class TestCheckbox
{ public static void main(String[] arg)
  { Frame f = new Frame("Optional veicolo");
    Panel p = new Panel();
    Checkbox ve_chk = new Checkbox("Vetri elettrici: ", true);
    Checkbox cc_chk = new Checkbox("Chiusura centralizzata: ", false);
    Checkbox af_chk = new Checkbox("Antifurto: ", false);
    Checkbox tc_chk = new Checkbox("Telecomando: ", false);
    Checkbox lt_chk = new Checkbox("Lunotto termico: ", false);
    p.add(ve_chk); p.add(cc_chk); p.add(af_chk); p.add(tc_chk);
    p.add(lt_chk); f.setLocation(100,100);
    f.setSize(300, 200); f.add(p); f.setVisible(true);
  }
} // end class
```

M. Malatesta 5.1.2 - Componenti-06

15
27/07/2011

5. Caselle di spunta (Checkbox)

Il listato dell'esempio precedente, crea un oggetto di classe **Checkbox** come quello mostrato in figura



M. Malatesta 5.1.2 - Componenti-06

16
27/07/2011

5. Caselle di spunta (Checkbox)

I seguenti sono i metodi maggiormente usati su oggetti di classe **Checkbox**.

METODO	EFFETTO
<code>Checkbox ident = new Checkbox();</code>	Crea una casella <i>ident</i> vuota
<code>Checkbox ident = new Checkbox(String testo);</code>	Crea una casella <i>ident</i> con <i>testo</i>
<code>Checkbox ident = new Checkbox (String testo, boolean s);</code>	Crea una casella <i>ident</i> con etichetta <i>testo</i> selezionata (<i>s=true</i>) o deselezionata (<i>s=false</i>)
<code>String c.getLabel ();</code>	Restituisce il testo presente nell'etichetta di <i>c</i>
<code>c.setLabel (String testo);</code>	Imposta il <i>testo</i> nella casella <i>c</i>
<code>boolean c.getState ();</code>	Restituisce lo stato (<i>true</i> o <i>false</i>) della casella <i>c</i>
<code>c.setState (boolean s);</code>	Imposta lo stato selezionato (<i>s=true</i>) o deselezionato (<i>s=false</i>) della casella <i>c</i>

M. Malatesta 5.1.2 - Componenti-06

17
27/07/2011

6. Pulsanti di opzione (CheckboxGroup)

```
import java.awt.*;
public class TestCheckboxGroup
{
    public static void main(String[] arg)
    {
        Frame f = new Frame("Iscrizione anno di corso");
        Panel p = new Panel();
        CheckboxGroup group = new CheckboxGroup();
        Checkbox uno_chkgrp = new Checkbox("1° anno", true, group);
        Checkbox due_chkgrp = new Checkbox("2° anno", false, group);
        Checkbox tre_chkgrp = new Checkbox("3° anno", false, group);
        p.add(uno_chkgrp); p.add(due_chkgrp); p.add(tre_chkgrp);
        f.setLocation (100,100); f.setSize (300, 200); f.add(p);
        f.setVisible (true);
    }
} // end class
```

M. Malatesta 5.1.2 - Componenti-06

18
27/07/2011

6. Pulsanti di opzione (CheckboxGroup)

I metodi più comuni della classe **CheckboxGroup**.

METODO	EFFETTO
CheckboxGroup <i>ident</i> = new CheckboxGroup ();	Crea una casella <i>ident</i> vuota
Checkbox <i>ident</i> = new Checkbox (<i>String testo</i>);	Crea una casella <i>ident</i> con <i>testo</i>
Checkbox <i>ident</i> = new Checkbox (<i>String testo</i> , boolean <i>s</i>);	Crea una casella <i>ident</i> con etichetta <i>testo</i> selezionata (<i>s</i> = true) o deselezionata (<i>s</i> = false)
Checkbox <i>ident</i> = new Checkbox (<i>String testo</i> , boolean <i>s</i> , CheckboxGroup <i>g</i>);	Crea una casella <i>ident</i> con etichetta <i>testo</i> selezionata (<i>s</i> = true) o deselezionata (<i>s</i> = false) facente parte del gruppo <i>g</i> di classe CheckboxGroup
<i>String c</i> .getLabel ();	Restituisce il testo presente nell'etichetta di <i>c</i>
<i>c</i> .setLabel (<i>String testo</i>);	Imposta il <i>testo</i> nella casella <i>c</i>
boolean <i>c</i> .getState ();	Restituisce lo stato (true o false) della casella <i>c</i>
<i>c</i> .setState (boolean <i>s</i>);	Imposta lo stato selezionato (<i>s</i> = true) o deselezionato (<i>s</i> = false) della casella <i>c</i>
<i>c</i> .getSelectedCheckbox()	Dà la checkbox correntemente impostata per il gruppo <i>c</i>
<i>c</i> .setSelectedCheckbox(Checkbox <i>box</i>)	Imposta come selezionata la checkbox <i>box</i> del gruppo <i>c</i>

M. Malatesta 5.1.2 - Componenti-06

19
27/07/2011

7. Caselle combinate (Choice)

```
import java.awt.*;
public class TestChoice
{ public static void main(String[] arg)
  { Frame f = new Frame("Selezione nominativo");
    Panel p = new Panel();
    Choice nomi = new Choice();
    nomi.addItem("Bianchi");
    nomi.addItem("Celestini");
    nomi.addItem("Rossi");
    nomi.addItem("Verdi");
    p.add(nomi); f.setLocation (100,100); f.setSize (300, 200);
    f.add(p); f.setVisible (true);
  }
} // end class
```

M. Malatesta 5.1.2 - Componenti-06

20
27/07/2011

7. Caselle combinate (Choice)

La casella a discesa (*combobox*) prodotta dall'esempio è la seguente.



M. Malatesta 5.1.2 - Componenti-06

21
27/07/2011

7. Caselle combinate (Choice)

I metodi più usati della classe **Choice**

METODO	EFFETTO
Choice <i>ident</i> = new Choice ();	Crea una casella combinata <i>ident</i> vuota
<i>c</i> .addItem (String <i>testo</i>);	Aggiunge la voce <i>testo</i> alla casella <i>c</i>
String getItem (int <i>index</i>)	Restituisce la voce in posizione <i>index</i>
int getItemCount ()	Restituisce il numero di voci presenti
int getSelectedItem ()	Restituisce la posizione della voce corrente
String getSelectedItem ()	Restituisce la voce corrente
insert (String <i>s</i> , int <i>index</i>)	Inserisce la voce <i>s</i> in posizione <i>index</i>
remove (int <i>index</i>)	Rimuove la voce in posizione <i>index</i>
remove (String <i>s</i>)	Rimuove la prima occorrenza della voce <i>s</i>
removeAll ()	Rimuove tutte le voci dal menu
select (int <i>index</i>)	Seleziona la voce in posizione <i>index</i>
select (String <i>s</i>)	Seleziona la voce <i>s</i>

M. Malatesta 5.1.2 - Componenti-06

22
27/07/2011

8. Liste (List)

```
import java.awt.*;
class TestList
{ final static int ITEMS = 10;
  static public void main(String[] args)
  { Frame f = new Frame("Esempio Listbox");
    List l = new List (ITEMS, true);
    for (int i = 0; i < ITEMS; i++)
    { l.add("item "+i);
      if (i%2==0) l.select(i);
    }
    f.add(l, BorderLayout.CENTER);
    f.pack();
    f.setVisible(true);
  }
}
```



M. Malatesta 5.1.2 - Componenti-06

23
27/07/2011

8. Liste (List)

I metodi più frequentemente usati della classe **List**

METODO	EFFETTO
List l = new List();	Crea una nuova lista <i>ident</i>
List l = new List(int rows);	Crea una lista <i>ident</i> con <i>rows</i> righe visibili
List l = new List(int rows, boolean m);	Crea una lista <i>ident</i> di <i>rows</i> righe in modo multiplo
void add (String item);	Aggiunge <i>item</i> in coda alla lista
void add (String item, int index);	Aggiunge <i>item</i> in posizione <i>index</i>
String getItem (int index);	Dà la stringa in posizione <i>index</i>
int getItemCount();	Dà il numero di elementi nella lista
int getRows();	Dà il numero di righe visibili della lista
int getSelectedIndex();	Dà la posizione dell'elemento selezionato
String getSelectedItem();	Dà l'elemento selezionato
boolean isIndexSelected();	Vero se elemento <i>index</i> è selezionato
boolean isMultipleMode();	Vero se la lista ammette selezioni multiple
void remove (int index)	Elimina elemento in posizione <i>index</i>
void removeAll()	Elimina tutti gli elementi dalla lista
void replaceItem (String item int index)	Rimpiazza elemento <i>index</i> con <i>item</i>
void deselect (int index)	Deseleziona elemento <i>index</i>
void select (int index)	Seleziona elemento <i>index</i>
void setMultipleMode()	Imposta il modo multiplo

M. Malatesta 5.1.2 - Componenti-06

24
27/07/2011

9. Barre di scorrimento (Scrollbar)

E' uno strumento che consente di scorrere un elenco o di selezionare un *range* di valori prestabiliti, mediante un cursore che può essere spostato dall'utente.

Le *scrollbar* hanno diverse proprietà, alcune delle quali sono:

- **orientamento** (orizzontale o verticale)
- **valore visibile** (la grandezza del cursore rispetto al range)
- **valore minimo** (il valore minimo rappresentabile)
- **valore massimo** (il valore massimo rappresentabile)
- **incremento unitario** (la minima unità di spostamento del cursore)
- **incremento a blocchi** (un multiplo dell'incremento unitario)

M. Malatesta 5.1.2 - Componenti-06

25
27/07/2011

9. Barre di scorrimento (Scrollbar)

```
import java.awt.event.*;
class TestScrollbar extends Frame
{ Scrollbar sbVer=new Scrollbar(Scrollbar.VERTICAL;
  Scrollbar sbHor=new Scrollbar(Scrollbar.HORIZONTAL,
                                0, 50, 10, 100); // valore, visibile, min, max

  TestScrollbar()
  { super("Esempio di Scrollbar");
    add(sbVer, BorderLayout.EAST);
    add(sbHor, BorderLayout.SOUTH);
    System.out.println("Scrollbar verticale"); displayScrollbar(sbVer);
    System.out.println("Scrollbar orizzontale"); displayScrollbar(sbHor);
    setSize(200, 200); setLocation(200, 300); setVisible(true);
  }
}
```

M. Malatesta 5.1.2 - Componenti-06

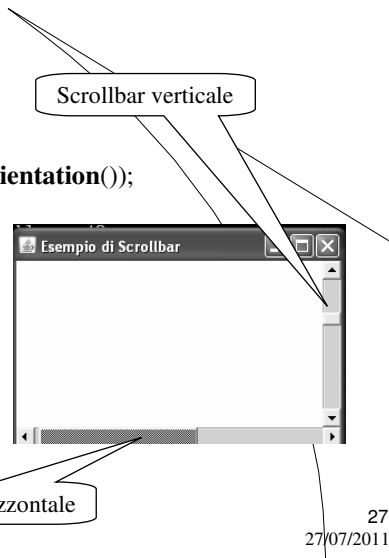
26
27/07/2011

9. Barre di scorrimento (Scrollbar)

```
static public void main(String[] args)
{
    new TestScrollbar();
}

void displayScrollbar (Scrollbar s)
{
    System.out.println("Orient: " + s.getOrientation());
    System.out.println("Incr. unitario: " +
        s.getUnitIncrement());
    System.out.println("Incr. blocco: " +
        s.getBlockIncrement());
    System.out.println("Visibile: " +
        s.getVisibleAmount());
}

} // end class
```



M. Malatesta 5.1.2 - Componenti-06

27
27/07/2011

9. Barre di scorrimento (Scrollbar)

I metodi più usati della classe **Scrollbar**

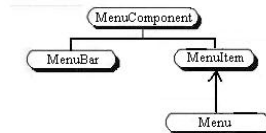
METODO	EFFETTO
<code>Scrollbar ident = new Scrollbar ();</code>	Crea una scrollbar ident
<code>Scrollbar ident=new Scrollbar (int direct);</code>	Crea una scrollbar ident orientata secondo direz
<code>Scrollbar ident=new Scrollbar (int direct, int value, int visible, int min, int max);</code>	Crea una scrollbar ident con le proprietà indicate
<code>int getBlockIncrement();</code>	Dà l'incremento a blocchi
<code>int getMaximum();</code>	Dà il valore massimo della barra
<code>int getMinimum();</code>	Dà il valore minimo della barra
<code>int getOrientation();</code>	Dà l'orientazione della barra
<code>int getUnitIncrement();</code>	Dà l'incremento unitario
<code>int getValue();</code>	Dà il valore corrente della barra
<code>int getVisibleAmount();</code>	Dà la porzione visibile del cursore
<code>int setBlockIncrement (int b);</code>	Imposta incremento a blocchi
<code>int setMaximum (int max);</code>	Imposta il massimo valore
<code>int setMinimum (int min);</code>	Imposta il minimo valore
<code>int setOrientation (int direct);</code>	Imposta l'orientazione
<code>int setUnitIncrement (int u);</code>	Imposta incremento unitario
<code>void setValue (int value);</code>	Imposta il valore della barra
<code>int setVisibleAmount (int visible);</code>	Imposta la porzione visibile del cursore

M. Malatesta 5.1.2 - Componenti-06

28
27/07/2011

10. Menu (MenuComponent)

Una parte della gerarchia delle classi, detta **MenuComponent**, è mostrato in figura ed è derivata direttamente da **Object**,



Queste classi ci consentono di creare oggetti che verranno visualizzati ed utilizzati come **menu a barra**. Lo schema di progettazione è il seguente:

- creare il menu a barra
- creare le voci del menu
- creare le opzioni di ogni singola voce
- impostare il menu nel contenitore

M. Malatesta 5.1.2 - Componenti-06

29
27/07/2011

10. Menu (MenuComponent)

```

import java.awt.*;
import java.util.*;
class TestMenu extends Frame
{ TestMenu()
{ super("Esempio di Menu a più livelli");
  setSize(100, 100); setLocation(200,200); show();
  MenuBar mb = new MenuBar();
  Menu m1 = new Menu("M1"); Menu m2=new Menu("M2");
  m1.add("M1-1"); m1.add("M1-2"); mb.add(m1);
  Menu p1=new Menu("M2-1");
  Menu p2=new Menu("M2-2");
  p1.add("M21-1"); p1.add("M21-2"); p2.add("M22-1");
  p2.add("M22-2"); p2.add("M22-3");
  m2.add(p1); m2.add(p2); mb.add(m2);
  setMenuBar(mb); // imposta il menu a barra sul frame
  menuDisplay(m1); menuDisplay(m2);
  menuDisplay(p1); menuDisplay(p2);
}
}
  
```

M. Malatesta 5.1.2 - Componenti-06

30
27/07/2011

10. Menu (MenuComponent)

```
public void menuDisplay(Menu m)
{ System.out.println("Menu "+
  m.getLabel() + " " + m.getItemCount() + " opzioni");
  System.out.println("Elenco oggetti:");
  for (int i=0; i<m.getItemCount(); i++)
  { System.out.println("Item " + i + ":" + m.getItem(i)); }
  System.out.println("Elenco opzioni:");
  for (int i=0; i<m.getItemCount(); i++)
  { MenuItem mi=(MenuItem)m.getItem(i);
    System.out.println(mi.getLabel());
  }
}
static public void main(String[] args)
{ new TestMenu(); }
} // end class
```

Visualizzazione del menu

Visualizzazione
delle singole voci

M. Malatesta 5.1.2 - Componenti-06

31
27/07/2011

10. Menu (MenuComponent)

I metodi più frequentemente usati della classe **Menu**.

METODO	EFFETTO
void add(String label)	Aggiunge <i>label</i> al menu
MenuItem getItem(int index)	Dà l'oggetto MenuItem in posizione <i>index</i>
int getItemCount()	Dà il numero di voci nel menu
void insert(String label, int index)	Inserisce nuova <i>label</i> in posizione <i>index</i>
void remove(int index)	Rimuove la voce in posizione <i>index</i>
void removeAll()	Rimuove tutte le voci dal menu

M. Malatesta 5.1.2 - Componenti-06

32
27/07/2011

Argomenti

- Componenti
 1. Etichette
 2. Pulsanti
 3. Caselle di testo
 4. Area di testo
 5. Caselle di spunta
 6. Pulsanti di opzione
 7. Caselle combinate
 8. Liste
 9. Barre di scorrimento
 10. Menu

M. Malatesta 5.1.2 - Componenti-06

33
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

M. Malatesta 5.1.2 - Componenti-06

34
27/07/2011