

## Una galleria di immagini con Java

```
/**
```

```
* Immagini.java da http://www.mrwebmaster.it/java/articoli/galleria-immagini-java\_754.html
```

```
*
```

```
* Immagini application
```

```
* una galleria di immagini con recupero di URL
```

```
*
```

```
* @author quarte
```

```
* @version 1.00 2011/4/9
```

```
*/
```

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

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

```
import javax.swing.*;
```

```
/*
```

```
* la mini gallery necessita di alcune immagini
```

```
* salvate nell'omonima cartella con lo stesso percorso dei file .class
```

```
*     immagini/rosso.gif
```

```
*     immagini/blu.gif
```

```
*     immagini/verde.gif
```

```
*     immagini/rosa.gif
```

```
*     immagini/giallo.gif
```

```
*     immagini/azzurro.gif
```

```
*     immagini/viola.gif
```

```
*/
```

```
public class Immagini extends JPanel implements ActionListener {
```

```
    //definiamone attributi e metodi
```

```
    JLabel pic;
```

```
    public Immagini() {
```

```
        super(new BorderLayout()); // per rendere possibili allineamenti [*]
```

```
                                //definiamo i nomi da associare alle immagini
```

```
        String[] coloPics = {
```

```
            "Rosso", "Blu", "Verde", "Rosa", "Giallo", "Azzurro", "Viola" };
```

```
        JComboBox coloList = new JComboBox(coloPics); //creiamo la lista delle immagini
```

```
                                // senza Generics
```

```
        coloList.setSelectedIndex(6); // si seleziona l'ultima voce
```

```
        coloList.addActionListener(this);
```

```
                                //definiamo le informazioni relative alle immagini
```

```
        pic = new JLabel();
```

```
        pic.setFont(pic.getFont().deriveFont(Font.BOLD));
```

```
        pic.setHorizontalAlignment(JLabel.LEFT); // allineamento [*]
```

```
        updateLabel(coloList.getSelectedIndex());
```

```
        pic.setBorder(BorderFactory.createEmptyBorder(5, 0, 0, 0));
```

```
                                //impostiamo la dimensione delle immagini
```

```
        pic.setPreferredSize(new Dimension(200, 110+10));
```

```
        add(coloList, BorderLayout.PAGE_START); //settiamo i bordi e i colori
```

```
        add(pic, BorderLayout.PAGE_END);
```

```
        setBorder(BorderFactory.createEmptyBorder(15, 15, 15, 15));
```

```
}
```



```

// gestione eventi di azione
public void actionPerformed(ActionEvent e) {
    JComboBox cb = (JComboBox)e.getSource();
    String nomePic = (String)cb.getSelectedItem();
    updateLabel(nomePic);
}

// recupero immagine con controllo esistenza file
// apertura e lettura del contenuto della cartella
protected void updateLabel(String name) {
    ImageIcon icon = createImageIcon("immagini/" + name + ".gif");
    pic.setIcon(icon);
    pic.setToolTipText("colore: " + name.toLowerCase());
    if (icon != null) {
        pic.setText(null);
    }else{
        pic.setText("Impossibile trovare la pagina");
    }
}

//notifica nel caso l'immagine non sia nel percorso specificato
protected static ImageIcon createImageIcon(String path) {
    java.net.URL imgURL = Immagini.class.getResource(path);
    //nomeapplicazione..class.getResource(path) per recuperare l'URL cioè il percorso del file
    if (imgURL != null) {
        return new ImageIcon(imgURL);
    }else{
        System.err.println("Impossibile trovare il file: " + path);
        return null;
    }
}

//creazione dell'interfaccia
private static void InterfacciaGrafica() {

                                                                    //setaggio della finestra

    JFrame frame = new JFrame("MiniGallery");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    JComponent newContentPane = new Immagini();
    newContentPane.setOpaque(true);
    frame.setContentPane(newContentPane);

                                                                    //mostriamo la finestra che contiene la GUI

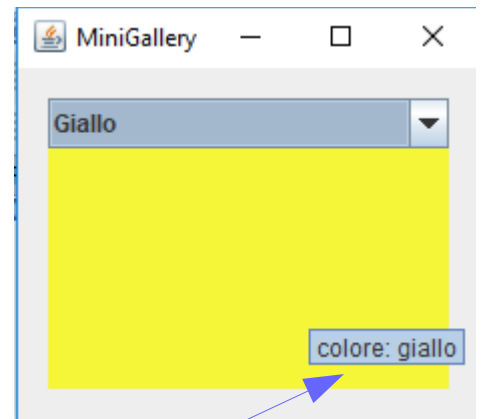
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {

                                                                    // creiamo il visualizzatore e mostriamo la GUI

    javax.swing.SwingUtilities.invokeLater(new Runnable() { public void run() {
        InterfacciaGrafica();}});
} // in realtà sarebbe sufficiente mostrare la GUI senza creare oggetto runnable
} //fine applicazione

```



**nb: da compilare con javac -Xlint**