APPENDICE: Gestione semplificata delle immagini

Le immagini memorizzate in file locali o su Internet possono essere lette in un'applicazione Java e visualizzate da oggetti Graphics. A partire dal JDK 1.4 la lettura di un'immagine è molto semplice e non richiede il recupero del "toolkit di default" nè la creazione di un oggetto di tipo MediaTracker per tracciare le immagini nel componente assegnato né l'attesa se più di un'immagine deve essere caricata prima di poterla disegnare.

Se l'immagine è memorizzata in un file locale si possono usare le seguenti linee di codice:

String filename= "*nome del file comprensivo del percorso*"
Image image = **ImageIO.read(new File(**filename**));**

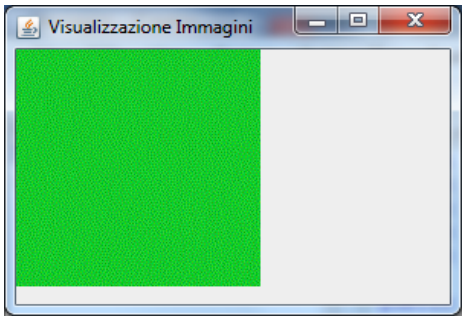
In alternativa è possibile indicare un URL:

String urlname= "*protocollo completo con nome risorsa*"
Image image = **ImageIO.read(new URL(**urlname**));**

Se l'immagine non è disponibile, il metodo read genera un'eccezione IOException.

A questo punto la variabile image contiene un riferimento ad un oggetto che incapsula i dati dell'immagine.

È possibile visualizzare l'immagine col metodo **drawImage** della classe Graphics.

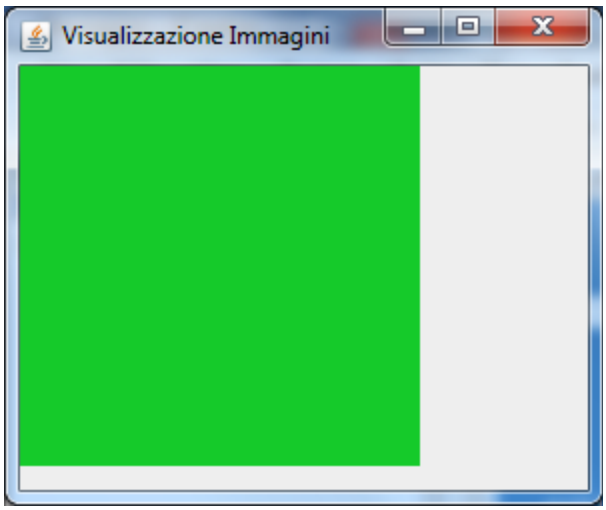


// Disegnare immagini su un pannello

```
import java.awt.*;
import java.awt.image.*;
import java.io.*;
import javax.imageio.ImageIO;
import javax.swing.*;

public class ImmaginePannello extends JPanel{

      private BufferedImage image;     // con attenzione alla bufferizzazione

      public ImmaginePannello() {
            try {
                  image = ImageIO.read(new File("immagini/verde.gif"));
                                                // cartella immagini in sottocartella con bytecode

             } catch (IOException e) {// handle exception. }
      }

      public void paintComponent(Graphics g) {

             g.drawImage(image, 0, 0, null);
             }

       public static void main(String [] args){

            JFrame f = new JFrame("Visualizzazione Immagini");
            ImmaginePannello p = new ImmaginePannello();
            f.setContentPane(p);
            f.setSize(300,210);
            f.setVisible(true);
            f.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
         }
 }
```

// Disegnare su un pannello immagini recuperate da Internet

```
import java.awt.*;
import java.awt.image.*;
import java.io.*;
import java.net.URL;
import javax.imageio.ImageIO;
import javax.swing.*;

public class ImmagineURL extends JPanel{

 private BufferedImage image;

 public ImmagineURL() {

   try {
         image = ImageIO.read(new URL
         ("http://www.disi.unige.it/person/MoggiE/LI03/lez6/images/verde.gif"));

     } catch (IOException ex) {
                                    // handle exception
                                  }

 }

 public void paintComponent(Graphics g) {

             g.drawImage(image, 0, 0, null);
             }

 public static void main(String [] args){

      JFrame f = new JFrame("Visualizzazione Immagini");
      ImmagineURL p = new ImmagineURL();
      f.setContentPane(p);
      f.setSize(300,250);
      f.setVisible(true);
      f.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
   }

 }
```



drawImage è un metodo con più forme, la più usata:

```
boolean Graphics.drawImage (Image img,                  //  l'immagine
                            int x, int y,               // coordinate dell'angolo a sinistra in alto
                            ImageObserver observer);    //  osservatore di immagini: oggetto in grado di controllare
                                                        //  un'immagine mentre viene caricata
                                                        //  (riceve la notifica del caricamento)
```