

Disegnare su JPanel

Esercizio: realizzare un'applicazione con un disegno di auto stilizzata usando componenti/contenitori *swing* inserendo il disegno su un pannello personalizzato (classe che eredita da JPanel)

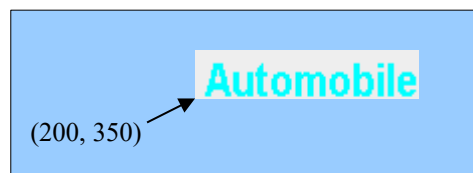
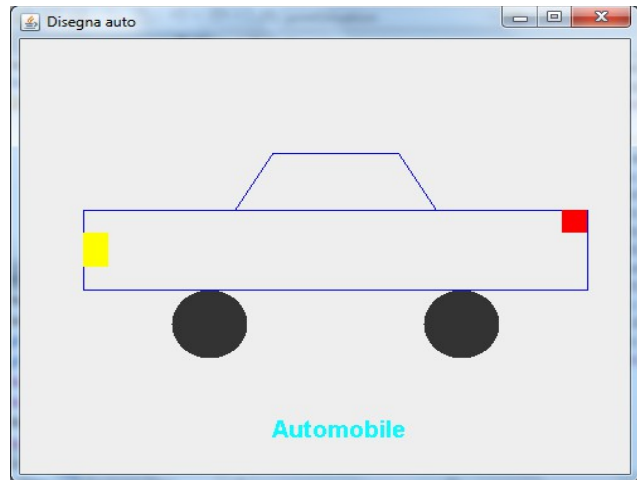
```
/**
 *
 * Auto application
 *
 * @author classi
 * @version 1.00 2011/3/25
 */
import java.awt.*;
import javax.swing.*;

public class Auto extends JFrame {
    private MioPanel A;
    private Container c;

    public Auto() {
        super ("Disegna auto");
        A = new MioPanel(); // istanza che eredita da JPanel
        c = getContentPane();
        c.add(A);
        setSize (500, 420);
        setLocation (50,50);
        setResizable(true); // con mouse ridimensionabile
        setVisible(true);
        setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {
        Auto o = new Auto();
    }
} // fine applicazione

// classe esterna
class MioPanel extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Font f = new Font("Times Roman", Font.BOLD, 20);
        g.setFont(f);
        g.fillOval(120, 220, 60,60); // una ruota piena
        g.fillOval(320, 220, 60,60); // altra ruota piena
        g.setColor(Color.blue); // carrozzeria blu
        g.drawRect(50, 150, 400, 70);
        g.drawLine(170,150,200,100);
        g.drawLine(330,150,300,100);
        g.drawLine(300,100,200,100);
        g.setColor(Color.yellow); // luci gialle
        g.fillRect(50, 170, 20, 30);
        g.setColor(Color.red); // luci rosse
        g.fillRect(430, 150, 20, 20);
        g.setColor(Color.cyan); // testo cyan
        g.drawString("Automobile", 200, 350);
    }
}
```



terminologia:

Un **frame** (lett.: 'cornice') è, a livello visivo, una superficie rettangolare provvista di una barra del titolo contenente, appunto, un titolo, un'icona e i tasti per la riduzione ad icona e per la chiusura della stessa; offre, inoltre, la possibilità di ridimensionare la superficie visibile mediante il mouse.

Nb:

Sopra a tutti i componenti di tipo **JComponent** (tra cui **JPanel**) si può disegnare: possiedono il metodo **paintComponent()** che non può essere chiamato direttamente ma :

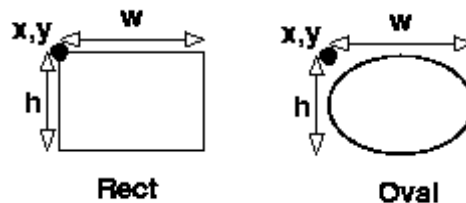
- automaticamente, quando si verifica un **evento** che provoca un **ridisegno della finestra** (ridimensionamento, riduzione a icona, visualizzazione etc...)
- esplicitamente con metodo **repaint()**

Nella **firma** del **metodo** possiamo notare la presenza di un **oggetto** particolare: **Graphics**.

Un **Graphics**, o **contesto grafico**, mette a disposizione una serie di metodi atti a scrivere, disegnare e inserire immagini sull'oggetto cui si riferisce. All'atto della creazione di un **JComponent**, quindi, viene associato ad esso un **Graphics**.

Metodi usati per il **disegno diretto** (con parametri interi esprimenti numero di pixel):

- void **draw/fillRect**(x,y, width,height): Disegna **rettangolo** vuoto o pieno.
- draw/**fillOval**(x,y, width,height): Disegna **ellisse**, vuoto o pieno.



- **drawLine**(x1,y1, x2,y2): Disegna **segmento di retta** tra (x1,y1) e (x2,y2).
- **drawString** (stringa,x,y): Disegna **stringa** all'interno di un rettangolo con vertice in basso a sinistra di coordinate (x,y). La stringa si estende a destra di x e sopra y, cioè per ascisse maggiori di x e ordinate MINORI di y.

Tutorial Sun <http://download.oracle.com/javase/tutorial/2d/TOC.html>

nb: l'origine delle coordinate nelle rappresentazioni grafiche sul monitor del PC è il vertice in alto a sinistra e l'asse delle ordinate è orientato come in figura

