



## Disegna Rettangolo

```
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import javax.swing.JComponent;
/** This component displays a rectangle that can be moved. */
public class RectangleComponent extends JComponent
{
    private static final int BOX_X = 100;
    private static final int BOX_Y = 100;
    private static final int BOX_WIDTH = 20;
    private static final int BOX_HEIGHT = 30;
    private Rectangle box;
    public RectangleComponent() {
        box = new Rectangle(BOX_X, BOX_Y, BOX_WIDTH, BOX_HEIGHT);
    }
    public void paintComponent(Graphics g)
    {
        Graphics2D g2 = (Graphics2D) g;
        g2.draw(box);
    }
}
```



## Sposta Rettangolo

```
/** Moves the rectangle to the given location.
@param x the x-position of the new location
@param y the y-position of the new location */
public void moveTo(int x, int y)
{
    box.setLocation(x, y);
    repaint();
}
/** Moves the rectangle by a given amount.
@param x the amount to move in the x-direction
@param y the amount to move in the y-direction */
public void moveBy(int dx, int dy)
{
    box.translate(dx, dy);
    repaint();
}
} //end of class
```



## Muovere Rettangolo

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame;
import javax.swing.Timer;

public class RectangleMover
{
    private static final int FRAME_WIDTH = 300;
    private static final int FRAME_HEIGHT = 400;
    public static void main(String[] args)
    {
        JFrame frame = new JFrame();
        frame.setSize(FRAME_WIDTH, FRAME_HEIGHT);
        frame.setTitle("An animated rectangle");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        final RectangleComponent component =
            new RectangleComponent();
        frame.add(component);
        frame.setVisible(true);
    }
}
```



## Muovere Rettangolo 2

```
class MousePressListener implements MouseListener
{
    public void mousePressed(MouseEvent event)
    {
        int x = event.getX();
        int y = event.getY();
        component.moveTo(x, y);
    }
    // Do-nothing methods
    public void mouseReleased(MouseEvent event) {}
    public void mouseClicked(MouseEvent event) {}
    public void mouseEntered(MouseEvent event) {}
    public void mouseExited(MouseEvent event) {}
}

MouseListener listener = new MousePressListener();
component.addMouseListener(listener);

} // end of main
} // end of class
```



## per Animazione Rettangolo

```
class TimerListener implements ActionListener
{
    public void actionPerformed(ActionEvent event)
    {
        component.moveBy(1, 1);
    }
}
ActionListener listener = new TimerListener();
final int DELAY = 100; // millisecc between timer ticks
Timer t = new Timer(DELAY, listener);
t.start();

} //end of main
} // end of class
```