import java.applet.
import java.awt.
import java.awt.event.

/**
 * A JOIPanel displays a button and a message. Pushing the button changes the message.
 * This panel can be displayed either from an applet in a browser or by the JVM as an application.
 * 
 * @version 10
 */

public class JOIPanel extends Applet {

private static final String MESSAGE1 = "Java Outside In";
private static final String MESSAGE2 = "Java Inside Out";
private String currentMessage = MESSAGE1; // currently displayed

private Font font;                // for printing the message
private Button button;            // for changing messages

/**
* Equip this Panel with a Button and an associated ButtonListener, and set the font for the message.
*/

public void init()
{
// what this Panel looks like
button = new Button( "Press Me" );
this.add( button );
font = new Font("Garamond", Font.BOLD, 48);

// how this Panel behaves
button.addActionListener( new JOIButtonListener( this ) );
}

/**
* Method that responds when the ButtonListener sends a changeMessage message.
*/

public void changeMessage()
{
currentMessage = 
    currentMessage.equals(MESSAGE1) ? MESSAGE2 : MESSAGE1;
this.repaint();
}

/**
* Draw the current message on this Panel.
* (The button is already there.)
* 
* @param g an object encapsulating the graphics (e.g. pen)
*/

public void paint(Graphics g)
{
g.setColor(Color.black);
g.setFont(font);
g.drawString(currentMessage, 40, 75);
}

/**
* Ask the JVM to display this Panel.
*/

public static void main( String[] args )
{
Terminal t     = new Terminal();
Frame frame    = new Frame();
JOIPanel panel = new JOIPanel();
panel.init();
frame.add(panel);
frame.setSize(400,120);
frame.show();
t.readLine("Type return to close the window ... ");
}
}
import java.awt.event.*;

/**
 * A simple listener for responding to button presses.
 * It knows the Panel on which the button lives, and
 * responds to button events by sending a changeMessage()
 * message to the Panel.
 * When a user presses the button, the method is
 * called. The actionListener behavior that must be
 * implemented.
 */

public class JOIButtonListener implements ActionListener
{
    private JOIPanel panel;  // the Panel containing the Button

    /**
     * Construct the ButtonListener.
     *
     * @param panel the Panel on which this Button will act.
     */
    public JOIButtonListener( JOIPanel panel )
    {
        this.panel = panel;
    }

    /**
     * Defines the ActionListener behavior that must be implemented.
     *
     * When a user presses the button on which this button will act,
     * construct the ButtonListener.
     */
    public void actionPerformed( ActionEvent e )
    {
        panel.changeMessage();
    }
}