

```

1 // foj/10/fojapplet/JOIApplet.java
2 //
3 //
4 // Copyright 2003 Bill Campbell and Ethan Bolker
5
6 import java.applet.*;
7 import java.awt.*;
8 import java.awt.event.*;
9
10 /**
11  * A JOIApplet displays a button and a message.
12  * Pushing the button changes the message.
13  */
14 * This class provides both the panel and the listener for
15 * the button on the panel - a common GUI programming idiom.
16
17 * The panel can be displayed either from an applet
18 * in a browser or by the JVM as an application.
19
20 * @version 1.0
21 */
22 */
23
24 public class JOIApplet extends Applet implements ActionListener
25 {
26     private static final String MESSAGE1 = "Java Outside In";
27     private static final String MESSAGE2 = "Java Inside Out";
28     private String currentMessage = MESSAGE1; // currently displayed
29
30     private Font font; // for printing the message
31     private Button button; // for changing messages
32
33     /**
34      * Equip this Panel with a Button
35      * and an associated ActionListener, and
36      * set the font for the message.
37      */
38
39     public void init()
40     {
41         // what this Panel looks like
42         button = new Button( "Press Me" );
43         this.add( button );
44         font = new Font("Garamond", Font.BOLD, 48);
45
46         // how this Panel behaves
47         button.addActionListener( this );
48     }
49
50     /**
51      * Defines the ActionListener behavior that must be
52      * implemented.
53      *
54      * When a user pushes the Button that we're listening to,
55      * send a changeMessage() message to the Panel.
56

```

```

57     * @param e the "event" when the button is pressed.
58     */
59
60     public void actionPerformed( ActionEvent e )
61     {
62         currentMessage =
63             currentMessage.equals( MESSAGE1 ) ? MESSAGE2 : MESSAGE1;
64         this.repaint();
65     }
66
67     /**
68      * Draw the current message on this Panel.
69      *
70      * (The button is already there.)
71      *
72      * @param g an object encapsulating the graphics (e.g. pen)
73      * properties.
74      */
75
76     public void paint( Graphics g )
77     {
78         g.setColor( Color.black );
79         g.setFont( font );
80         g.drawString( currentMessage, 40, 75 );
81     }
82
83     /**
84      * Ask the JVM to display this Panel.
85      */
86
87     public static void main( String[] args )
88     {
89         Terminal t = new Terminal();
90         Frame frame = new Frame();
91         JOIApplet panel = new JOIApplet();
92         panel.init();
93         frame.add( panel );
94         frame.setSize( 400, 120 );
95         frame.show();
96         t.readLine( "Type return to close the window ... " );
97         System.exit( 0 );
98     }
99 }

```