

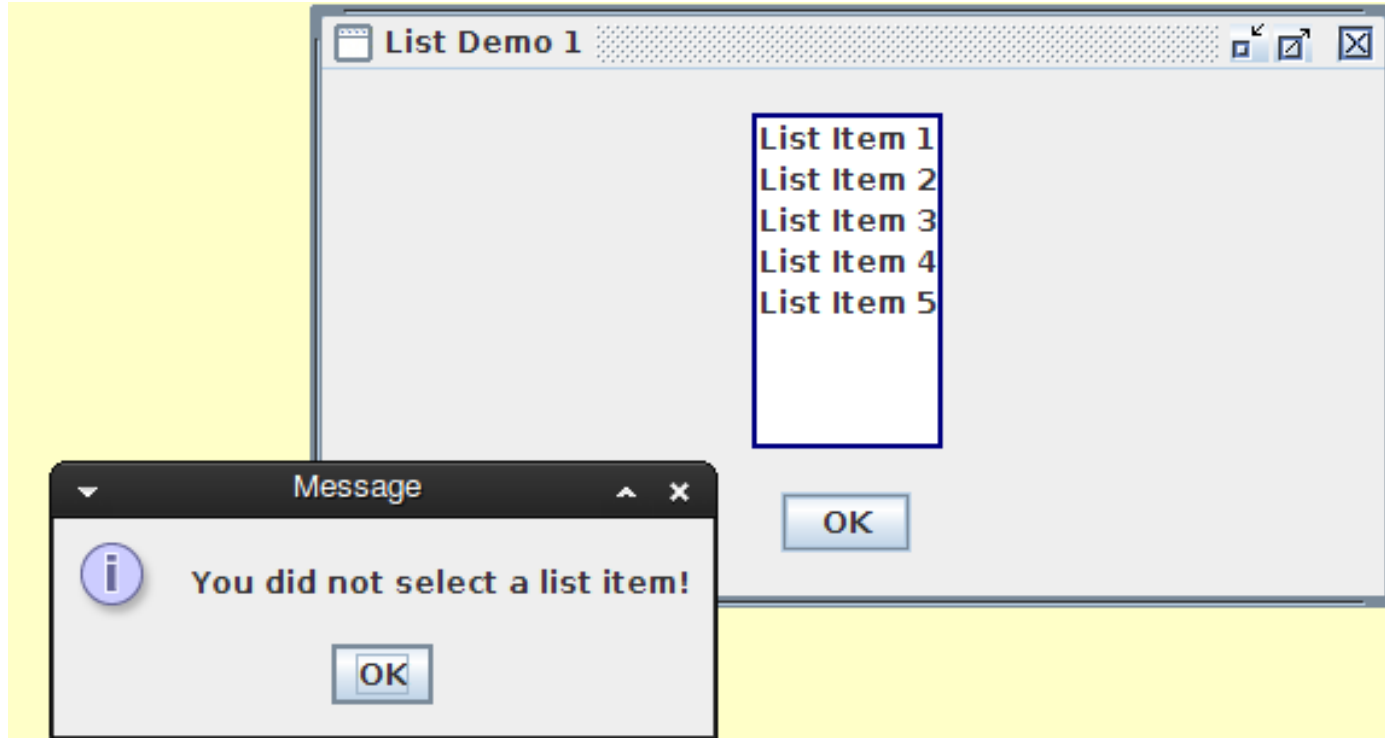
18. GUI Programming :: The *JList* Class (continued from Section 17)

We can determine which list item is selected by calling *JList.getSelectedIndex()*:

```
list.getSelectedIndex();
```

The items are numbered starting at 0. We can obtain the list item that was selected by calling *JList.getSelectedValue()*:

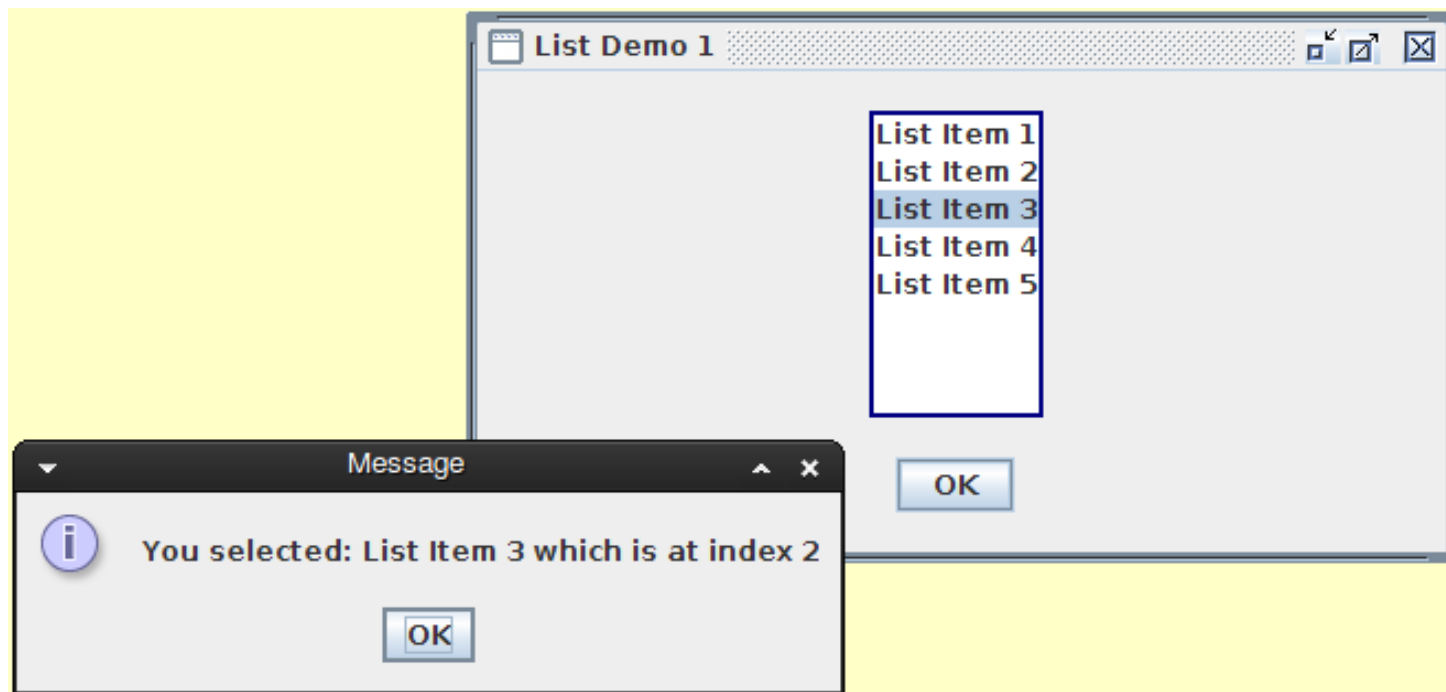
```
String selected = list.getSelectedValue();
```



18. GUI Programming :: The *JList* Class (continued)

If no list item was selected *getSelectedIndex()* returns -1 and *getSelectedValue()* returns null:

```
if (index == -1) {  
    msg = "You did not select a list item!";  
} else {  
    msg = "You selected: " + list.getSelectedValue();  
    msg += " which is at index " + index;  
}
```



18. GUI Programming :: The *JList* Class :: *ListDemo1*

```
//*****  
// CLASS: ListDemo1 (ListDemo1.java)  
//*****  
  
import java.awt.Color;  
import java.awt.event.ActionListener;  
import java.awt.event.ActionEvent;  
import javax.swing.BorderFactory;  
import javax.swing.Box;  
import javax.swing.BoxLayout;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JList;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.ListSelectionModel;  
  
/**  
 * This application demonstrates how to create a vertical list using the javax.  
 * swing.JList.  
 */  
  
public class ListDemo1 {  
    public static void main(String[] args) { new ListDemo1().run(); }  
    public void run() {  
        // Use the Swing look and feel.  
        JFrame.setDefaultLookAndFeelDecorated(true);
```

18. GUI Programming :: The *JList* Class :: *ListDemo1* (continued)

```
// Create an array of 5 strings to be added to the JList.  
String[] listItems = new String[5];  
for (int i = 0; i < 5; ++i) listItems[i] = "List Item " + (i + 1);  
  
// Create a vertical JList of Strings. Draw a 2 pixel-wide dark blue line  
// border around the list. setAlignment(0.5) specifies that the list  
// will be centered in the middle of its region within the BoxLayout.  
// list is declared as final so it can be accessed in the actionPerformed()  
// method of the butOk button listener.  
final JList<String> list = new JList<>(listItems);  
list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);  
list.setLayoutOrientation(JList.VERTICAL);  
list.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 128), 2));  
list.setAlignmentX(0.5f);
```

18. GUI Programming :: The *JList* Class :: *ListDemo1* (continued)

```
JButton butOk = new JButton("OK");
butOk.setAlignmentX(0.5f);
butOk.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent pEvent) {
            int index = list.getSelectedIndex();
            String msg;
            if (index == -1) {
                msg = "You did not select a list item!";
            } else {
                msg = "You selected: " + list.getSelectedValue();
                msg += " which is at index " + index;
            }
            JOptionPane.showMessageDialog(null, msg);
        }
    });

JPanel mainPanel = new JPanel();
mainPanel.setLayout(new BoxLayout(mainPanel, BoxLayout.Y_AXIS));
mainPanel.add(Box.createVerticalGlue());
mainPanel.add(list);
mainPanel.add(Box.createVerticalGlue());
mainPanel.add(butOk);
mainPanel.add(Box.createVerticalGlue());
```

18. GUI Programming :: The *JList* Class :: *ListDemo1* (continued)

```
JFrame frame = new JFrame("List Demo 1");
frame.setSize(450, 250);
frame.add(mainPanel);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
}
```