## The JList as an Observer

Now, what about that list box in our color changing example? We saved it for last because, as we noted in the Adapter class discussion (Chapter 9), the JList is rather different in concept than the List object in the AWT. You can display a fixed list of data in the JList by simply putting the data into a Vector or a String array. However, to display a list of data that might grow or otherwise change, you must put that data into a special data object derived from the AbstractListModel class and then use that class in the constructor to the JList class. The ListFrame class looks like this:

```
public class ListFrame extends JFrame
implements Observer {
    private JList          list;
    private  JPanel        p;
    private JScrollPane    lsp;
    private JListData      listData;

    public ListFrame(Subject s) {
        super("Color List");
        //put the panel into the frame
        p = new JPanel(true);
        getContentPane().add("Center", p);
        p.setLayout(new BorderLayout());
        //tell the Subject that we are interested
        s.registerInterest(this);

        //Create the list
        listData = new JListData();    //the list model
        list = new JList(listData);    //the visual list
        lsp = new JScrollPane();       //the scroller
        lsp.getViewport().add(list);
        p.add("Center", lsp);
        lsp.setPreferredSize(new Dimension(100,100));
        setBounds(250, 100, 100, 100);
        setVisible(true);
    }
    //--------------
    public void sendNotify(String s) {
        listData.addElement(s);
    }
}
```

We name the ListModel class JListData. It holds the Vector that contains the growing list of color names.

```
public class JListData extends AbstractListModel {
    private Vector data;
    public JListData() {
        data = new Vector();
```