13 April 2010 GUIs: Graphical User Interfaces

Read Chap. 17 of the text. ProgramLive CD: a better way to learn about GUIs. See the CD for examples of code.

Their mouse had a mean time between failure of ... week ... it would jam up irreparably, or ... jam up on the table-- ... it had a flimsy cord whose wires would break. Steve Jobs: "... Xerox says it can't be built for < $400, I want a $10 mouse that will never fail and can be mass produced, because it's going to be the primary interface of the computer ...

... Dean Hovey ... came back, "I've got some good and some bad news. Their mouse had a mean time between failure of … a week … it would jam up irreparably, or ... jam up on the table-- .....


Basic Components

Component  Button, Canvas, Checkbox, Choice, Label, List, ScrollBar, TextField, TextArea

ComponentExample(String t) {
  JFrameDemo.java

Putting components in a JFrame

import java.awt.*;
import java.swing. *;
/** Demonstrate placement of components in a JFrame. Use BorderLayout.
   It places five components in the five possible areas:
   (1) a JButton in the east,
   (2) a JLabel in the west,
   (3) a JLabel in the south,
   (4) a JTextField in the north, and
   (5) a JTextArea in the center. */
public class ComponentExample extends JFrame {
  /**
   * Class constructor: set the title and add five components
   */
  public ComponentExample(String t) {
    super(t);
    Container cp= getContentPane();
    cp.add(new JTextField("click me"), BorderLayout.EAST);
    cp.add(new JLabel("label 2"), BorderLayout.WEST);
    cp.add(new JLabel("label 3"), BorderLayout.SOUTH);
    cp.add(new JTextField("type here", 4, 10), BorderLayout.CENTER);
    cp.add(new JTextField("type here", 22), BorderLayout.NORTH);
    cp.add(new JButton("Click here"), BorderLayout.NORTH);
    cp.add(new JLabel("label 4"), BorderLayout.SOUTH);
    cp.add(new JLabel("label 5"), BorderLayout.EAST);
    cp.setVisible(true);
    pack();
  }
}

JFrame's content pane

Layout manager: An instance controls the placement of components.

JFrame layout manager default: BorderLayout.

BorderLayout layout manager: Can place 5 components:

<table>
<thead>
<tr>
<th>Component</th>
<th>JComponent</th>
<th>JPanel</th>
<th>JApplet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box</td>
<td>Container</td>
<td>Frame</td>
<td>JWindow</td>
</tr>
</tbody>
</table>

What components can go in a JFrame

Packages that contain classes that deal with GUIs:
java.awt: Old package. java.swing: New package.

JavaX.swing has a better way of listening to buttons, text fields, etc. Its components are more flexible.

<table>
<thead>
<tr>
<th>Component</th>
<th>Something that can be placed in a GUI window. They are instances of certain classes, e.g.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JButton</td>
<td>Clickable button</td>
</tr>
<tr>
<td>JLabel</td>
<td>Line of text</td>
</tr>
<tr>
<td>JTextField</td>
<td>Field into which the user can type</td>
</tr>
<tr>
<td>JTextArea</td>
<td>Many-row field into which user can type</td>
</tr>
</tbody>
</table>

Components that can contain other components

<table>
<thead>
<tr>
<th>Component</th>
<th>Box</th>
<th>Container</th>
<th>JComponent</th>
<th>JPanel</th>
<th>JApplet</th>
<th>JFrame</th>
<th>JWindow</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.awt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>java.swing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note the use of subclasses to provide structure and efficiency. For example, there are two kinds of JToggleButton, so that class has two subclasses.
**import** java.awt.*; **import** javax.swing.*;

/** Instance has labels in east/west, JPanel with four buttons in center. */
public class PanelDemo extends JFrame {
JPanel p=new JPanel();
/** Constructor: a frame with title "Panel demo", labels in east/west, blank label in south, Panel of 4 buttons in the center */
public PanelDemo() {
   super("Panel demo");
p.add(new JButton("0")); p.add(new JButton("1"));
p.add(new JButton("2")); p.add(new JButton("3"));
   Container cp= getContentPane();
   cp.add(new JLabel("east"), BorderLayout.EAST);
   cp.add(new JLabel("west"), BorderLayout.WEST);
   cp.add(new JLabel("    "), BorderLayout.SOUTH);
   cp.add(p, BorderLayout.CENTER);
   pack(); show();
}
}

**import** javax.swing.*; **import** java.awt.*;

/** Demo class Box. Comment on constructor says how frame is laid out. */
public class BoxDemo extends JFrame {
   Box b=new Box(BoxLayout.X_AXIS);
b.add(new JButton("0")); b.add(new JButton("1"));
b.add(new JButton("2")); b.add(new JButton("3"));
   Container cp= getContentPane();
   cp.add(new JLabel("east"), BorderLayout.EAST);
   cp.add(new JLabel("west"), BorderLayout.WEST);
   cp.add(new JLabel("    "), BorderLayout.SOUTH);
   cp.add(b, BorderLayout.CENTER);
   pack(); show();
}

public class BoxDemo2 extends JFrame {
/** Constructor: frame with title t and 3 columns with n, n+1, and n+2 buttons. */
public BoxDemo2(String t, int n) {
super(t);
// Create Box b1 with n buttons.
   Box b1=new Box(BoxLayout.Y_AXIS);
   for (int i=0; i<n; i++)
      b1.add(new JButton("1 " + i));
// Create Box b2 with n+1 buttons.
   Box b2=
// Create Box b3 with n+2 buttons.
   Box b3=
// Create horizontal box b containing b1, b2, b3
   Box b= new Box(BoxLayout.X_AXIS);
b.add(b1);
b.add(b2);
b.add(b3);
   Container cp= getContentPane();
   cp.add(b, BorderLayout.CENTER);
   pack(); show();
}

Boxes within a Box
3 vertical boxes, each a column of buttons, are placed in a horizontal box

Simulate BoxLayout Manager in a JFrame

To simulate using a BoxLayout manager for a JFrame, create a Box and place it as the sole component of the JFrame:

JFrame jf=new JFrame("title");
Box b=new Box(BoxLayout.X_AXIS);
Add components to b;
jf.add(b, BorderLayout.CENTER);

Interested in learning more about GUIs?
1. Start developing a GUI by changing an already existing one. There are a lot of details, and it is hard to get all the details right when one starts from scratch and has little idea about the Java GUI package.

2. The easiest way to learn about GUIs is to listen the ProgramLive lectures in Chapter 17. That chapter shows you code for everything, and you can also download the code from the CD and compile and use it yourself.

3. We have shown you how to place components in a GUI. We haven‘t yet shown you how to “listen” to things like button clicks in a GUI. That comes later.