12 Nov 2008 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 … a 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. Good news: we've got a new project with Apple. Bad news: I told Steve we'd design a mouse for 10 bucks."

... year later ... we … filed … and were granted a patent, on the electro-mechanical-optical mouse of today; ... we ended up ... [making] the mouse as invisible to people as it is today.

Steve Sachs interview on 1st computer with GUI: Apple Lisa (about $9,999 in 1982).
http://library.stanford.edu/mac/primary/interviews/sachs/trans.html

JFrame's content pane

Layout manager: An instance controls the placement of components.

JFrame layout manager default: BorderLayout.

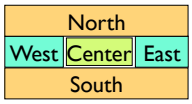
BorderLayout layout manager: Can place 5 components:

```
Container cp= getContentPane();

JButton jb= new JButton("Click here");
JLabel jl= new JLabel("label 2");

cp.add(jb, BorderLayout.EAST);
cp.add(jl, BorderLayout.WEST);

pack();
setVisible(true);
```



JFrameDemo.java

Putting components in a JFrame

```
import java.awt.*;
import javax.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 {
   /** Constructor: a window with title t and 5 components */
   public ComponentExample(String t) {
      super(t);
      Container cp=  getContentPane();
      cp.add(new JButton("click me"), BorderLayout.EAST);
      cp.add(new JTextField("type here", 22), BorderLayout.NORTH);
      cp.add(new JCheckBox("I voted today"), BorderLayout.SOUTH);
      cp.add(new JLabel("label 2"), BorderLayout.WEST);
      cp.add(new JTextArea("type\nhere", 4, 10), BorderLayout.CENTER);
      pack();
   }
```

ComponentExample.java

What components can go in a JFrame

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

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

Component: Something that can be placed in a GUI window. They are instances of certain classes, e.g.

JButton, Button: Clickable button
JLabel, Label: Line of text
JTextField, TextField: Field into which the user can type:
JTextArea, TextArea: Many-row field into which user can type
JPanel, Panel: Used for graphics; to contain other components
JCheckBox: Checkable box with a title
JComboBox: Menu of items, one of which can be checked
JRadioButton: Same functionality as JCheckBox
Container: Can contain other components
Box: Can contain other components

Basic Components

Component
  Button, Canvas
  Checkbox, Choice
  Label, List, Scrollbar
  TextComponent
    TextField, TextArea
  Container
    JComponent
    AbstractButton
      JButton
      JToggleButton
        JCheckBox
        RadioButton
    JLabel, JList
    JOptionPane, JPanel
    JPopupMenu, JScrollBar, JSlider
    JTextComponent
    JTextField, JTextArea

Component: Something that can be placed in a GUI window. These are the basic ones that one uses in a GUI

Note the use of subclasses to provide structure and efficiency. For example, there are two kinds of JToggleButtons, so that class has two subclasses.

Components that can contain other components

Component
  Box
  Container
    JComponent
    JPanel
    Panel
      Applet
  Window
    Frame
      JFrame
    JWindow

java.awt is the old GUI package.

javax.swing is the new GUI package. When they wanted to use an old name, they put J in front of it.

(e.g. Frame and JFrame)

When constructing javax.swing, the attempt was made to rely on the old package as much as possible.

So, JFrame is a subclass of Frame.

But they couldn't do this with JPanel.

```
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, JPanel 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();
   }
}
```

JPanel as a container

JPanel layout manager default: FlowLayout.

FlowLayout layout manager: Place any number of components. They appear in the order in which they were added, taking as many rows as necessary.

7

```
import javax.swing.*;  import java.awt.*;
/** Demo class Box. Comment on constructor says how frame is laid out.*/
public class BoxDemo extends JFrame {
   /** Constructor: frame with title "Box demo", labels in the east/west,
       blank label in south, horizontal Box with 4 buttons in center.*/
   public BoxDemo() {
      super("Box demo");
      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,
      pack(); show();
   }
}
```

Class Box: a container

Box layout manager default: BoxLayout.

BoxLayout layout manager: Place any number of components. They appear in the order in which they were added, taking only one row.

8

```
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= i+1)
            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

BoxLayout layout manager: Place any number of components. They appear in the order in which they were added, taking only one row.

9

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);
```

10

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.

11