

Java Swing

- ## AWT to Swing
- AWT: Abstract Windowing Toolkit
 - import java.awt.*
 - Swing: new with Java2
 - import javax.swing.*
 - Extends AWT
 - Many new improved components
 - Standard dialog boxes, tooltips, ...
 - Look-and-feel, skins
 - Event listeners
 - API:
 - <http://java.sun.com/j2se/1.3/docs/api/index.html>

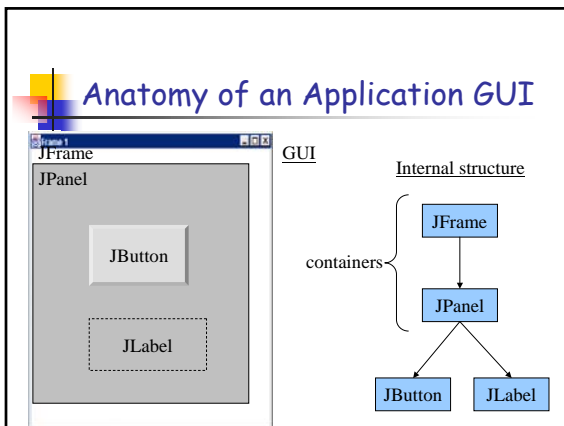
GUI Component API

- Java: GUI component = class
- Properties
- Methods
- Events

```

    graph LR
      subgraph API
        P[Properties]
        M[Methods]
        E[Events]
      end
      subgraph JButton
        B[JButton]
      end
      P --> B
      M --> B
      B --> E
    
```

- ## Using a GUI Component
1. Create it
 - Instantiate object: `b = new JButton("press me");`
 2. Configure it
 - Properties: `b.text = "press me";` [avoided in java]
 - Methods: `b.setText("press me");`
 3. Add it
 - `panel.add(b);`
 4. Listen to it
 - Events: Listeners
-



- ## Using a GUI Component 2
1. Create it
 2. Configure it
 3. Add children (if container)
 4. Add to parent (if not JFrame)
 5. Listen to it
- order important ↓

Build from bottom up

- Create:
 - Frame
 - Panel
 - Components
 - Listeners
- Add: (bottom up)
 - listeners into components
 - components into panel
 - panel into frame

```

graph TD
    Listener[Listener] --> JButton[JButton]
    JLabel[JLabel] --> JPanel[JPanel]
    JButton --> JPanel
    JPanel --> JFrame[JFrame]
  
```

Code

```

JFrame f = new JFrame("title");
JPanel p = new JPanel( );
JButton b = new JButton("press me");

p.add(b);           // add button to panel
f.setContentPane(p); // add panel to frame

f.show();
  
```

Application Code

```

import javax.swing.*;

class hello {
  public static void main(String[] args){
    JFrame f = new JFrame("title");
    JPanel p = new JPanel();
    JButton b = new JButton("press me");

    p.add(b);           // add button to panel
    f.setContentPane(p); // add panel to frame

    f.show();
  }
}
  
```

Layout Managers

- Automatically control placement of components in a panel
- Why?

Layout Manager Heuristics

null none, programmer sets x,y,w,h	FlowLayout Left to right Top to bottom	GridLayout
--	---	-----------------------

Layout Manager Heuristics

BorderLayout 	CardLayout One at a time 	GridBagLayout
-------------------------	--	--------------------------

Coordinate System

- Upside-down Cartesian

- $y_{window} = height - y_{cartesian}$

Component Hierarchy

- Each component has its own subwindow
 - Subwindow = rectangular area within component
 - Has own coordinate system
- Clipping:
 - Can't paint outside its subwindow
 - Can't paint over child components?

Combinations

Combinations

Code: null layout

```

JFrame f = new JFrame("title");
JPanel p = new JPanel( );
JButton b = new JButton("press me");

b.setBounds(new Rectangle(10,10, 100,50));
p.setLayout(null); // x,y layout
p.add(b);
f.setContentPane(p);
  
```

Code: FlowLayout

```

JFrame f = new JFrame("title");
JPanel p = new JPanel( );
FlowLayout L = new FlowLayout( );
JButton b1 = new JButton("press me");
JButton b2 = new JButton("then me");

p.setLayout(L);
p.add(b1);
p.add(b2);
f.setContentPane(p);
  
```

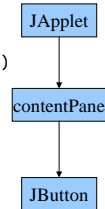
Set layout mgr before adding components

Applets

- JApplet is like a JFrame
- Already has a panel
 - Access panel with JApplet.getContentPane()

```
import javax.swing.*;

class hello extends JApplet {
    public void init(){
        JButton b = new JButton("press me");
        getContentPane().add(b);
    }
}
```



Applet Methods

- Called by browser:
 - init() - initialization
 - start() - resume processing (e.g. animations)
 - stop() - pause
 - destroy() - cleanup
 - paint() - redraw stuff ('expose' event)

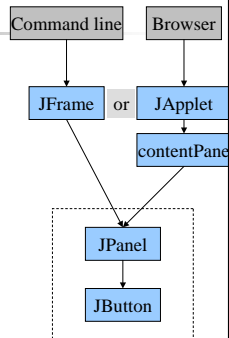
Application + Applet

```
import javax.swing.*;

class helloApp {
    public static void main(String[] args){
        // create JFrame and put my mainPanel in it
        JFrame f = new JFrame("title");
        mainPanel p = new mainPanel();
        f.setContentPane(p);
        f.show();
    }
}

class helloApplet extends JApplet {
    public void init(){
        // put my mainPanel in the Applet
        mainPanel p = new mainPanel();
        getContentPane().add(p);
    }
}

// my main GUI is in here:
class mainPanel extends JPanel {
    mainPanel(){
        setLayout(new FlowLayout());
        JButton b = new JButton("press me");
        add(b);
    }
}
```



Applet Security

- No read/write on client machine
- Cannot execute programs on client machine
- Communicate only with server
- "Java applet window" Warning