ITSE322 Modern Programming Language: Advanced Java

Java GUI

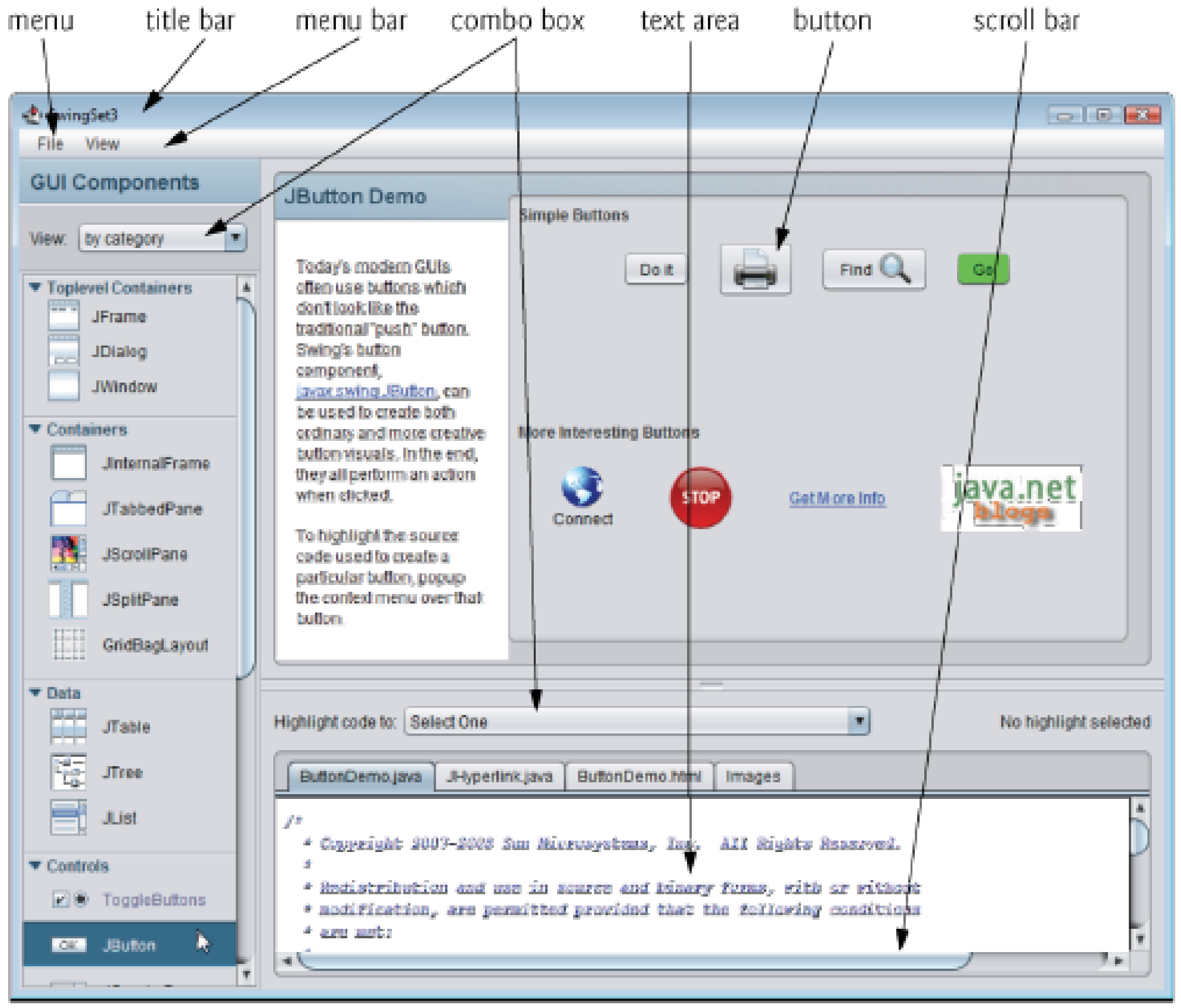
Lecture 5

Learning Objectives

1. Create simple graphical user interfaces (GUI’s) in Java
2. Learn about event-driven model
3. Build GUI for your database

What’s in a GUI?

Answer:
A bunch of graphical objects!



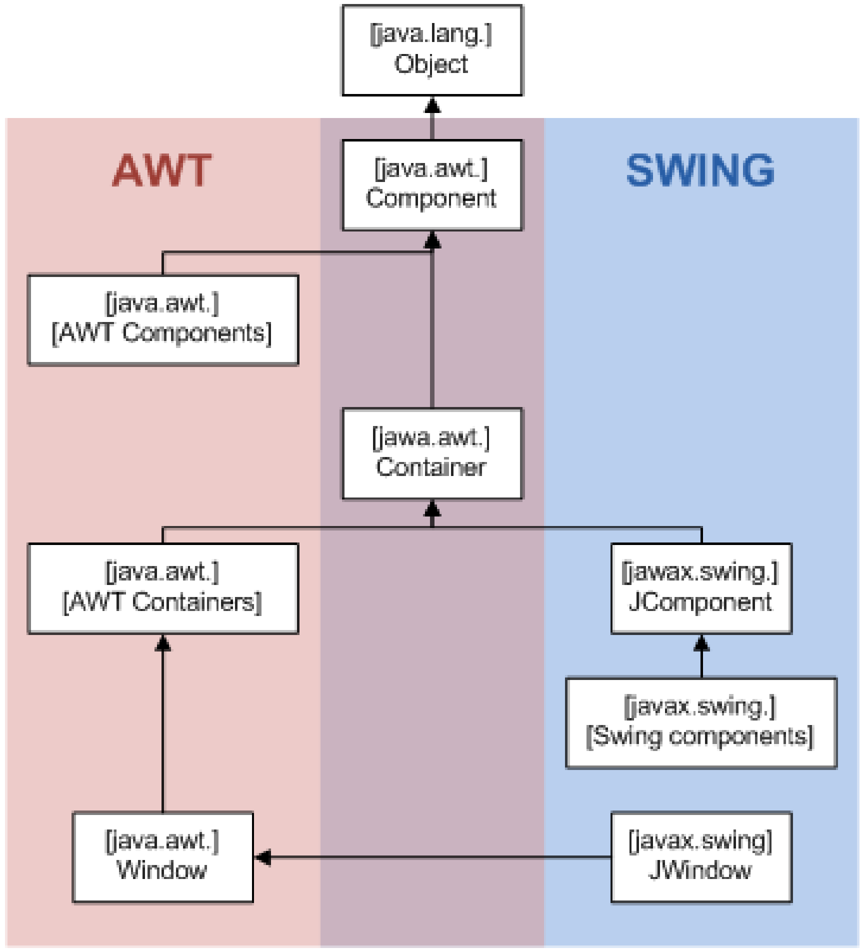
Today's modern GUIs often use buttons which don't look like the traditional "push" button. Swing's button component, javax.swing.JButton, can be used to create both ordinary and more creative button visuals. In the end, they all perform an action when clicked.

To highlight the source code used to create a particular button, popup the context menu over that button.

```
/*
 * Copyright 2007-2008 Sun Microsystems, Inc.  All Rights Reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
```

The Java GUI framework

- Abstract Windowing Toolkit (AWT)
  - Built on the native OS
  - Faster
  - Can be used in browsers without a java plugin
- Swing
  - Newer – built on AWT.
  - Made completely in Java
  - More Portable
  - Easier to use
  - Can use the 'Model View Control' design process

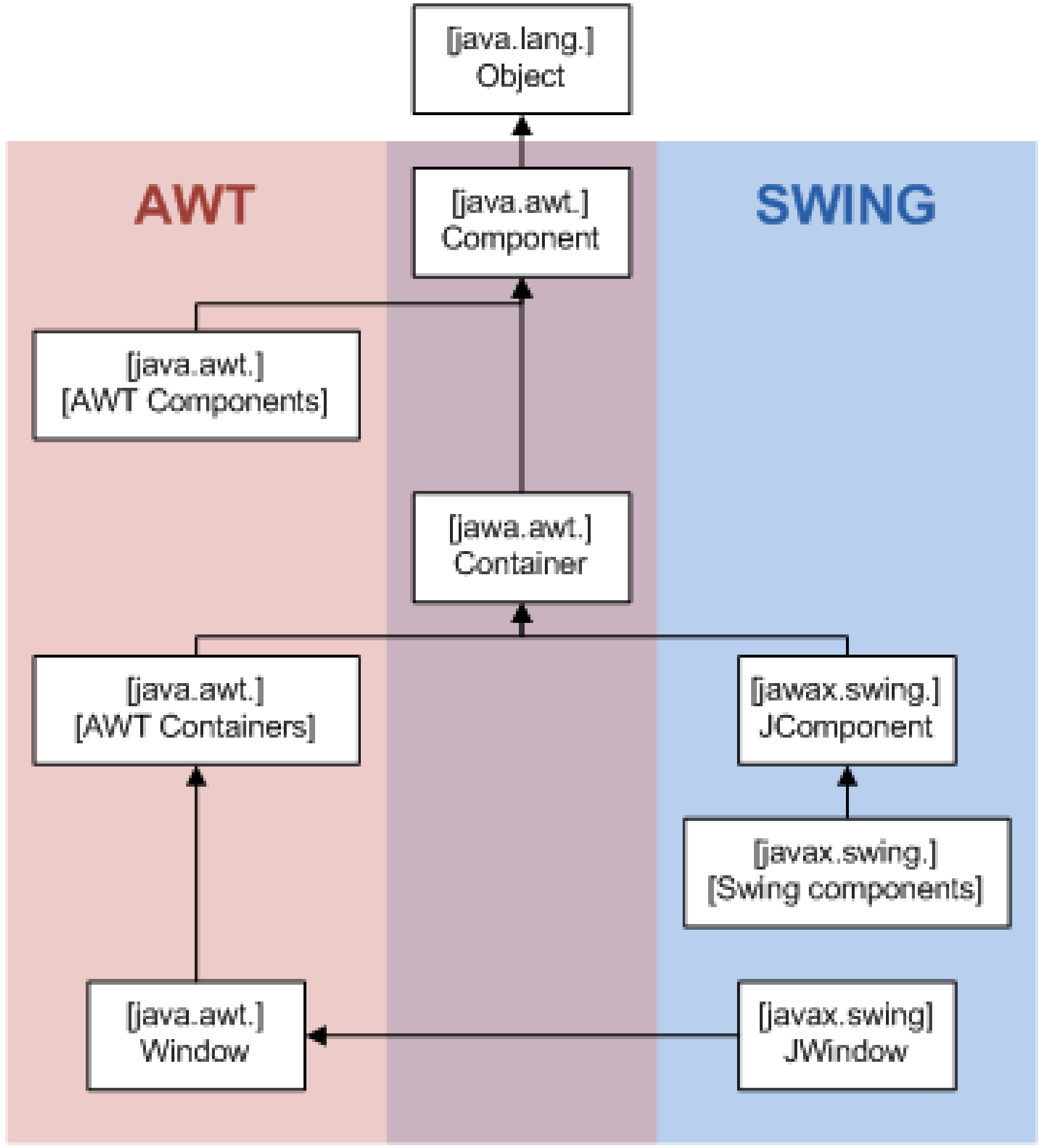

[java.lang.] Object
AWT
[java.awt.] Component
SWING
[java.awt.] [AWT Components]
[jawa.awt.] Container
[java.awt.] [AWT Containers]
[jawax.swing.] JComponent
[javax.swing.] [Swing components]
[java.awt.] Window
[javax.swing] JWindow

API:

http://java.sun.com/j2se/1.3/docs/api/index.html

Swing

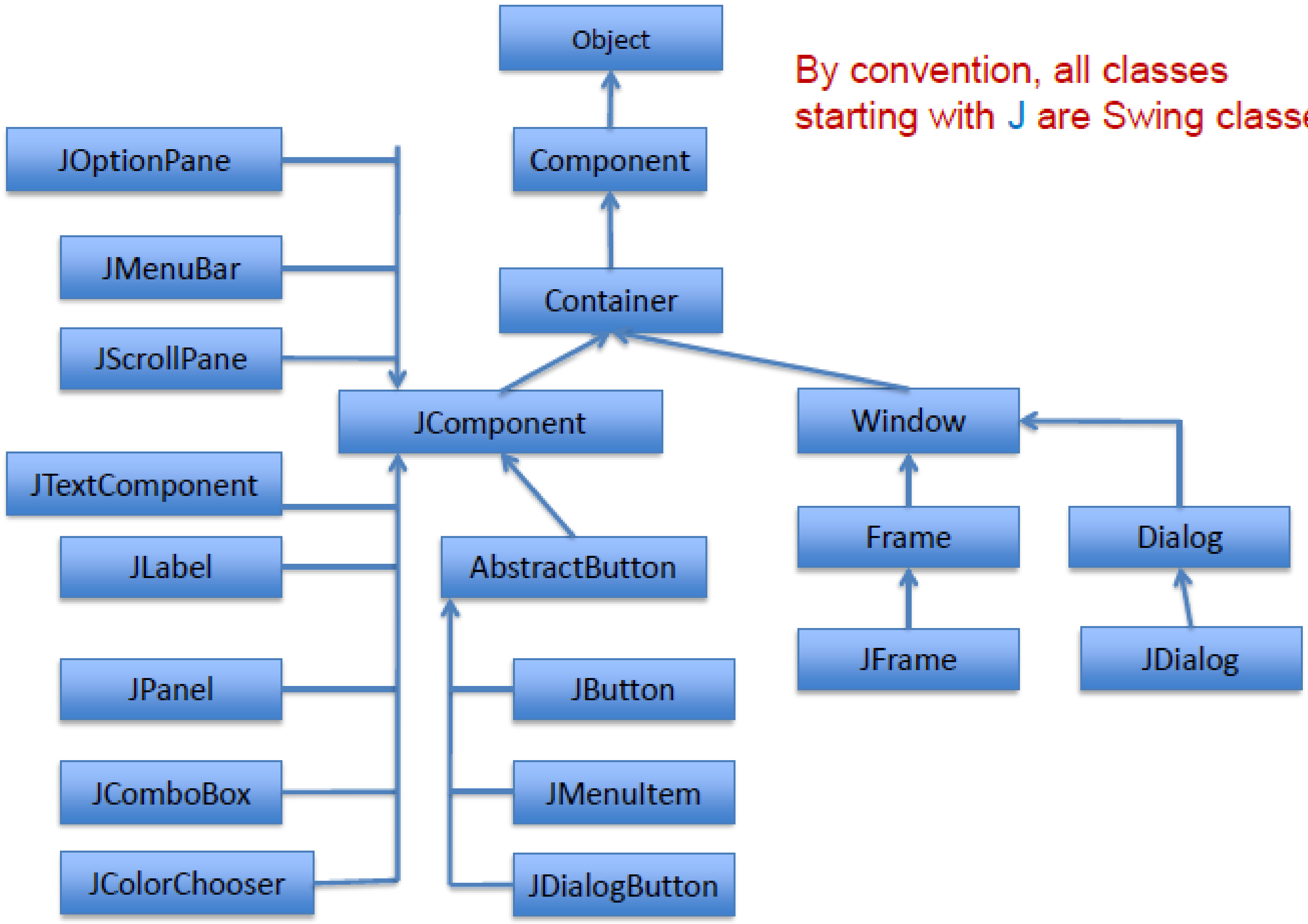
- The JComponent class is the root of the swing component hierarchy
    - All swing components are subtypes of this except for top-level containers such as JFrame


[java.lang.] Object
AWT
[java.awt.] Component
SWING
[java.awt.] [AWT Components]
[jawa.awt.] Container
[java.awt.] [AWT Containers]
[jawax.swing.] JComponent
[javax.swing.] [Swing components]
[java.awt.] Window
[javax.swing] JWindow

More Swing Components

Component	Description
`JLabel`	An area that can display text
`JTextField`	An area in which the user may type a single line of input from the key board
`JComboBox`	A component that displays a drop-down list of items from which the user may select. A combo box also provides a text field in which a use may type input. It is a combo box as it is a combination of a list and a text field
`JCheckBox`	A component that has a box that may be checked or unchecked
`List`	A list from which a user may select an item
`JRadioButton`	A control that can be either selected or deselected. Radio buttons usually appear in groups and allow the user to select one of several options
`JSlider`	A control that allows the user to select a value by moving a slider along a track
`JButton`	A button that can cause an action to occur when clicked

Hierarchical View



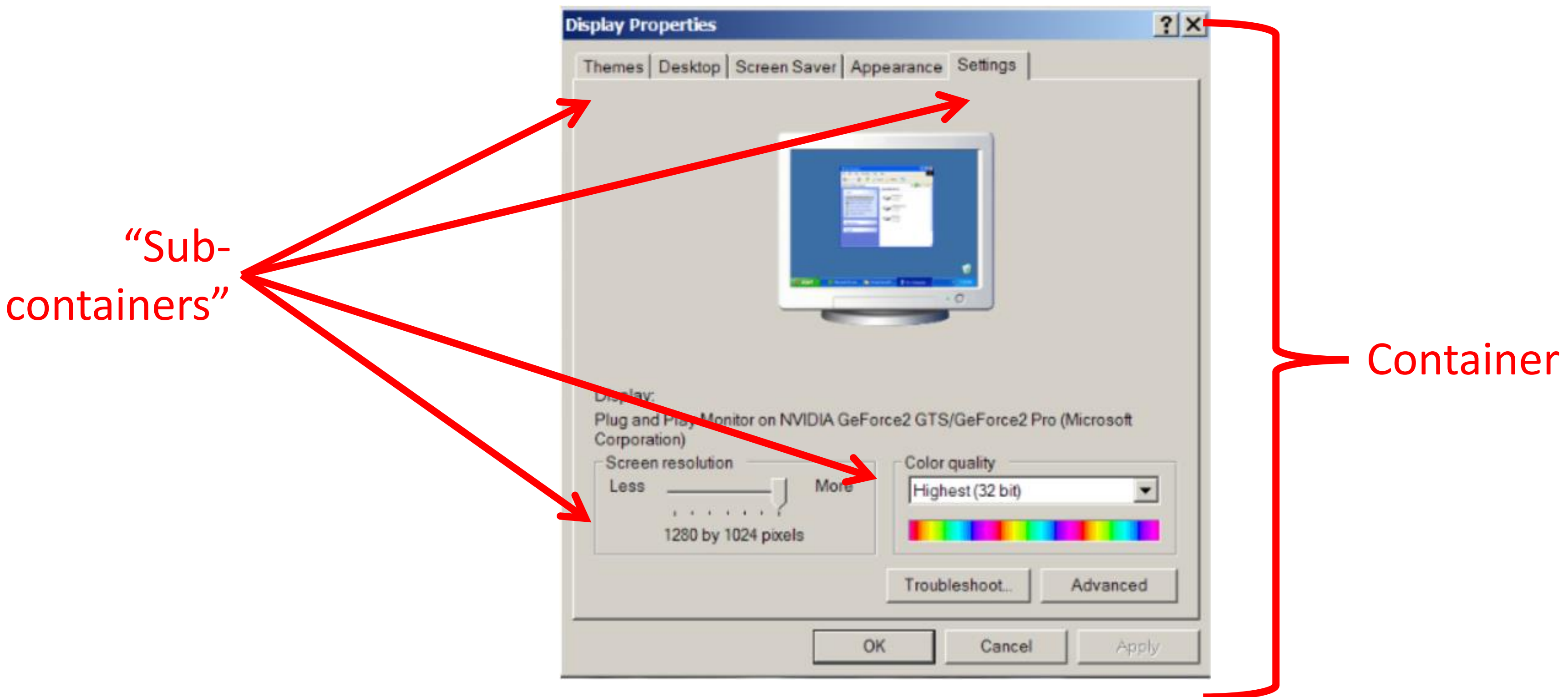
By convention, all classes starting with J are Swing classes

Components

- There are many types of graphical controls and displays available:
  - `JButton, JFrame, JLabel, JList, JTextArea, Window`
- A graphical component is also known as a “widget”

Containers

- A special type of `Component` that is used to hold other components.
- Can be used to group components on the screen (i.e., one container holds another container which in turn groups a number of controls).

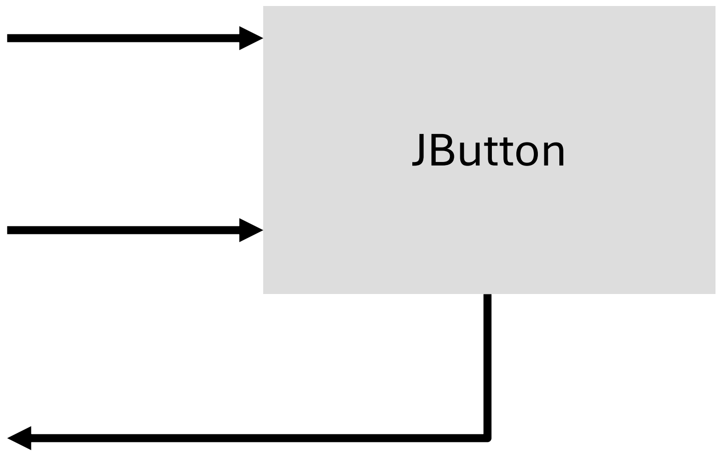


Display:
Plug and Play Monitor on NVIDIA GeForce2 GTS/GeForce2 Pro (Microsoft Corporation)
Screen resolution: 1280 by 1024 pixels
Color quality: Highest (32 bit)

GUI Component API

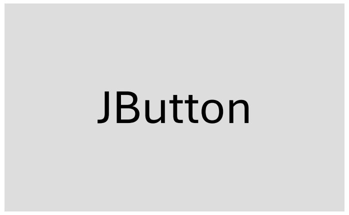
- Java: GUI component = class

- Properties
    -
- Methods
    -
- Events
    -

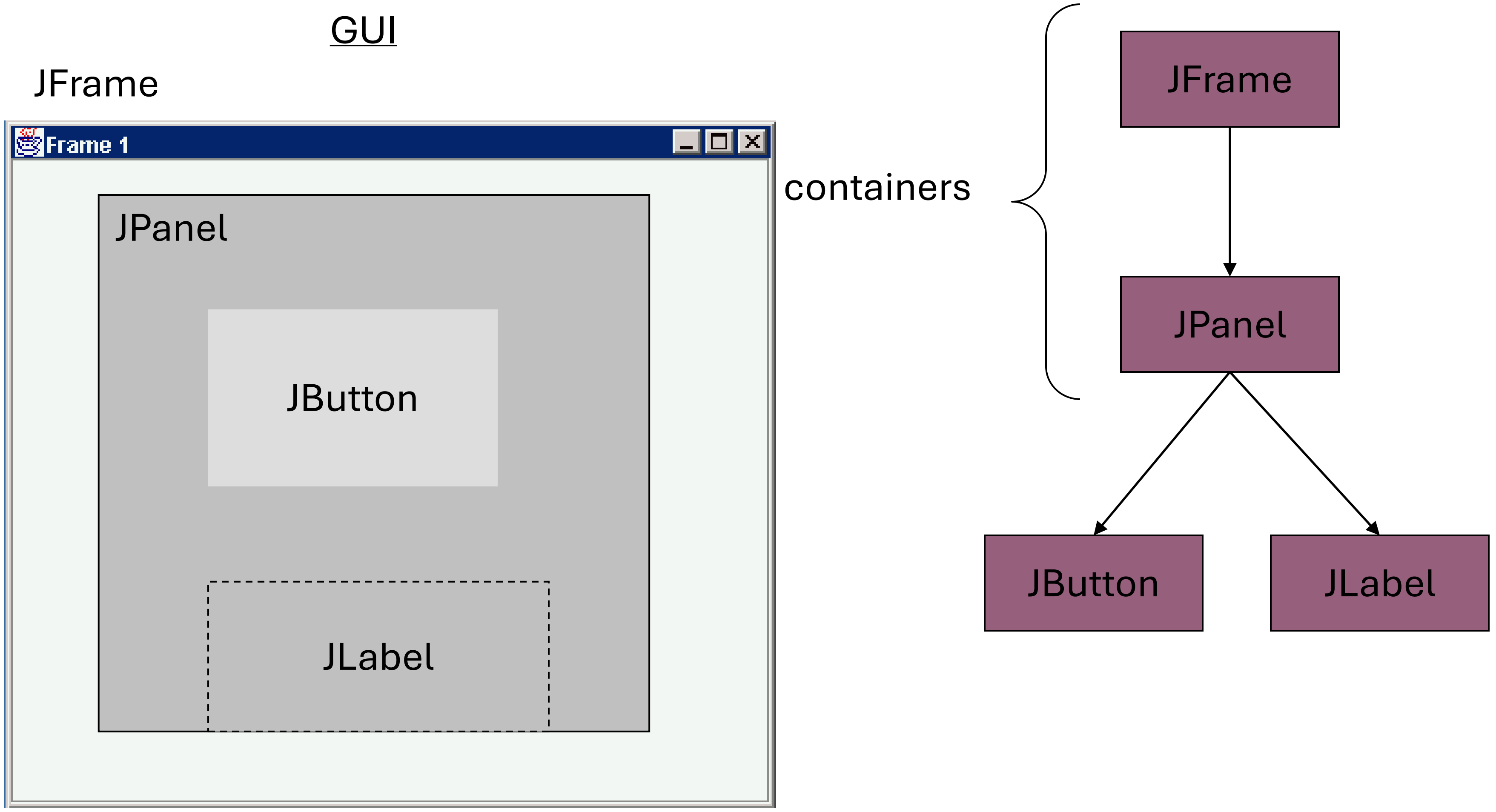


Using a GUI Component

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

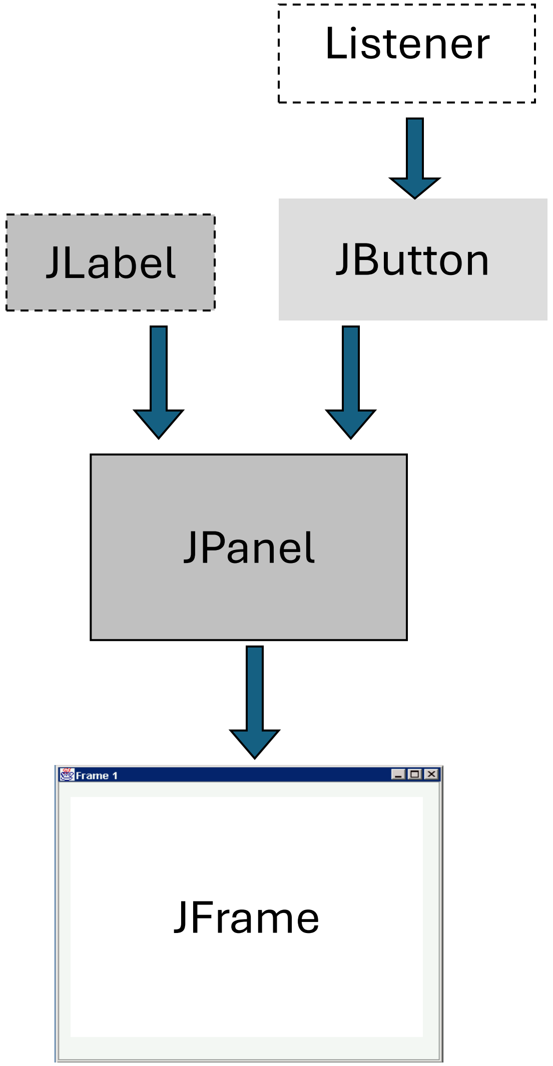


Anatomy of an Application GUI



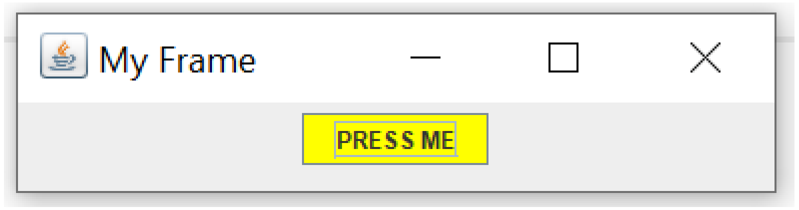
Build from bottom up

- Create:
  - Frame
  - Panel
  - Components
  - Listeners
- Add:
  - listeners into components
  - components into panel
  - panel into frame



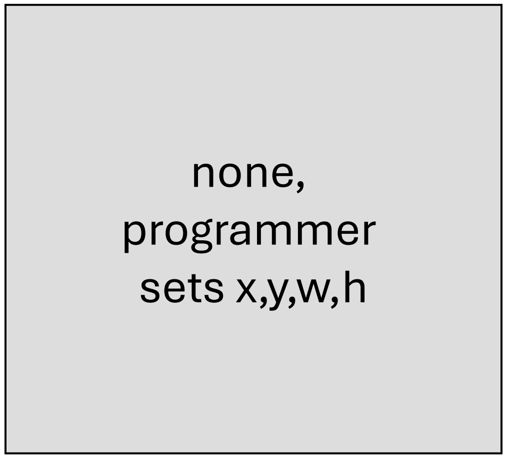
Code

```java
import java.awt.Color;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class SimpleGUI1 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("TITLE");
        //1. Create it
        JPanel panel = new JPanel();
        JButton button = new JButton("PRESS ME");
        //2. Configure it
        frame.setTitle("My Frame");
        frame.setSize(400,100);
        button.setBackground(Color.YELLOW);
        //3. add it
        panel.add(button); // add button to panel
        frame.setContentPane(panel); // add panel to frame
        frame.setVisible(true);
    }
}
```

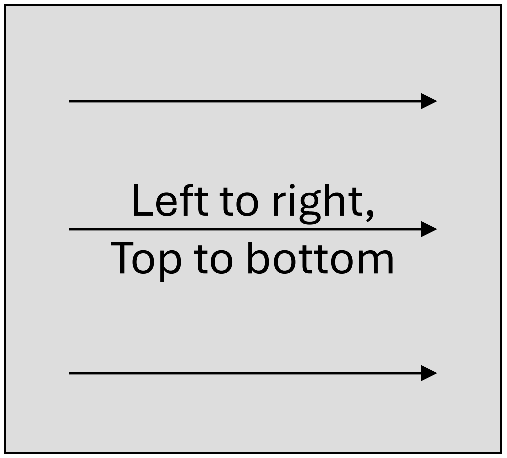


Layout Manager Heuristics

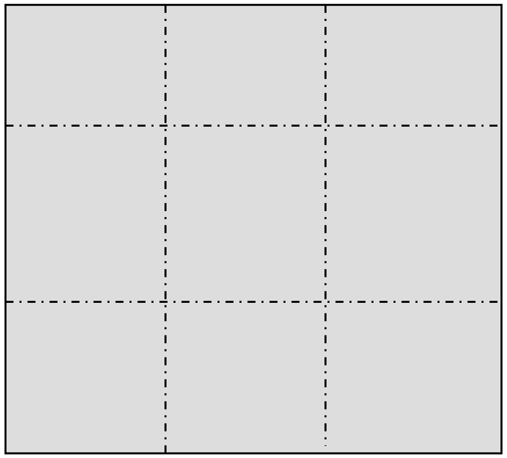
null



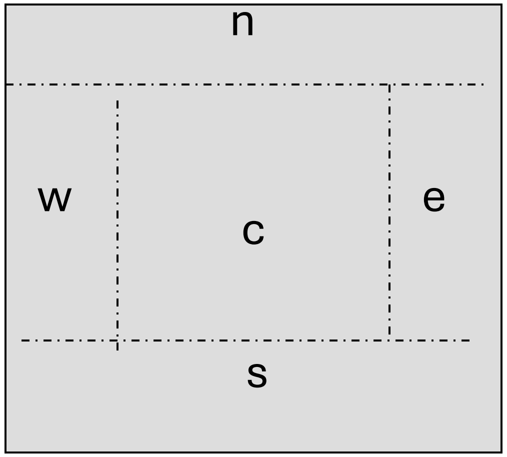
FlowLayout



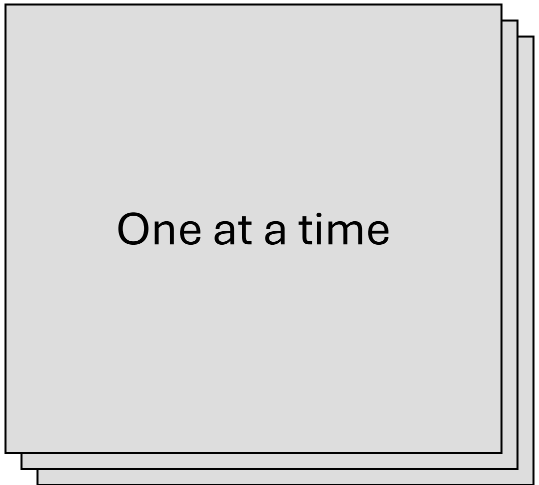
GridLayout



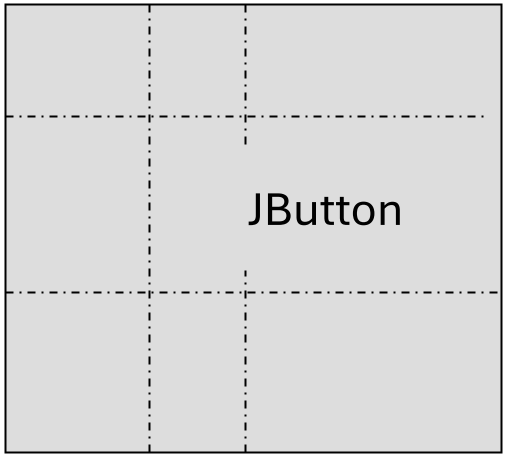
BorderLayout



CardLayout

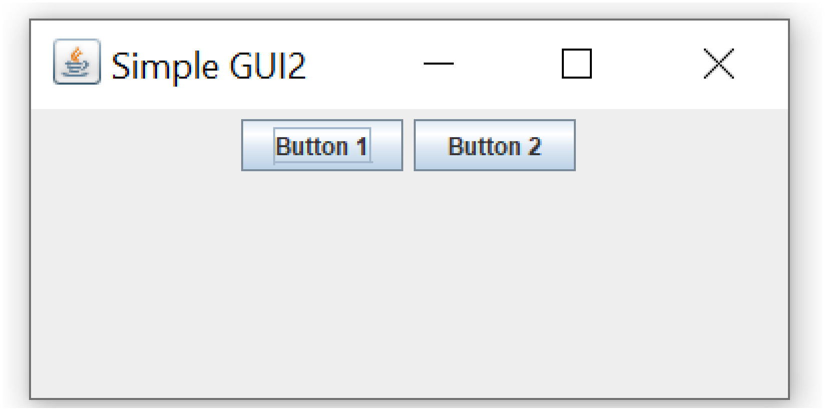


GridBagLayout



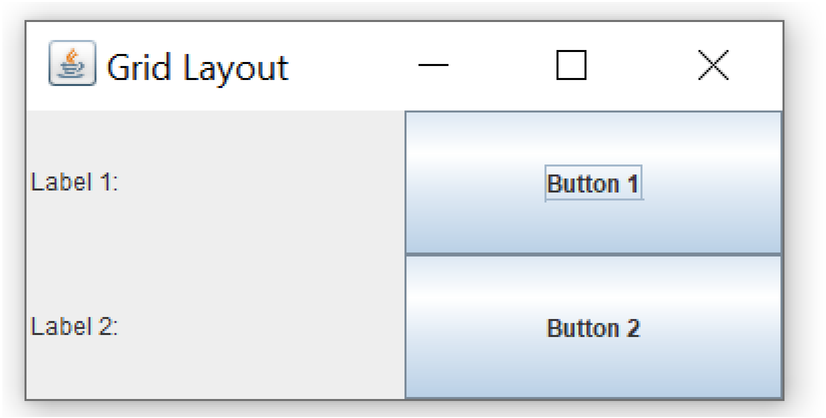
Flow Layout

```
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.Container;
import java.awt.FlowLayout;
class SimpleGUI2 extends JFrame{
   public SimpleGUI2()
   {
        setDefaultCloseOperation(EXIT_ON_CLOSE);
         //add button
         JButton but1 = new JButton("Button 1");
         JButton but2 = new JButton("Button 2");
        Container cp = getContentPane();//must do this
        cp.setLayout(new FlowLayout());
         cp.add(but1);
         cp.add(but2);
         setTitle("Simple GUI2");
        setVisible(true);
    }
     public static void main(String[] args)
     {
        SimpleGUI2 gui = new SimpleGUI2();
        gui.setSize(400,200);
     }
}
```

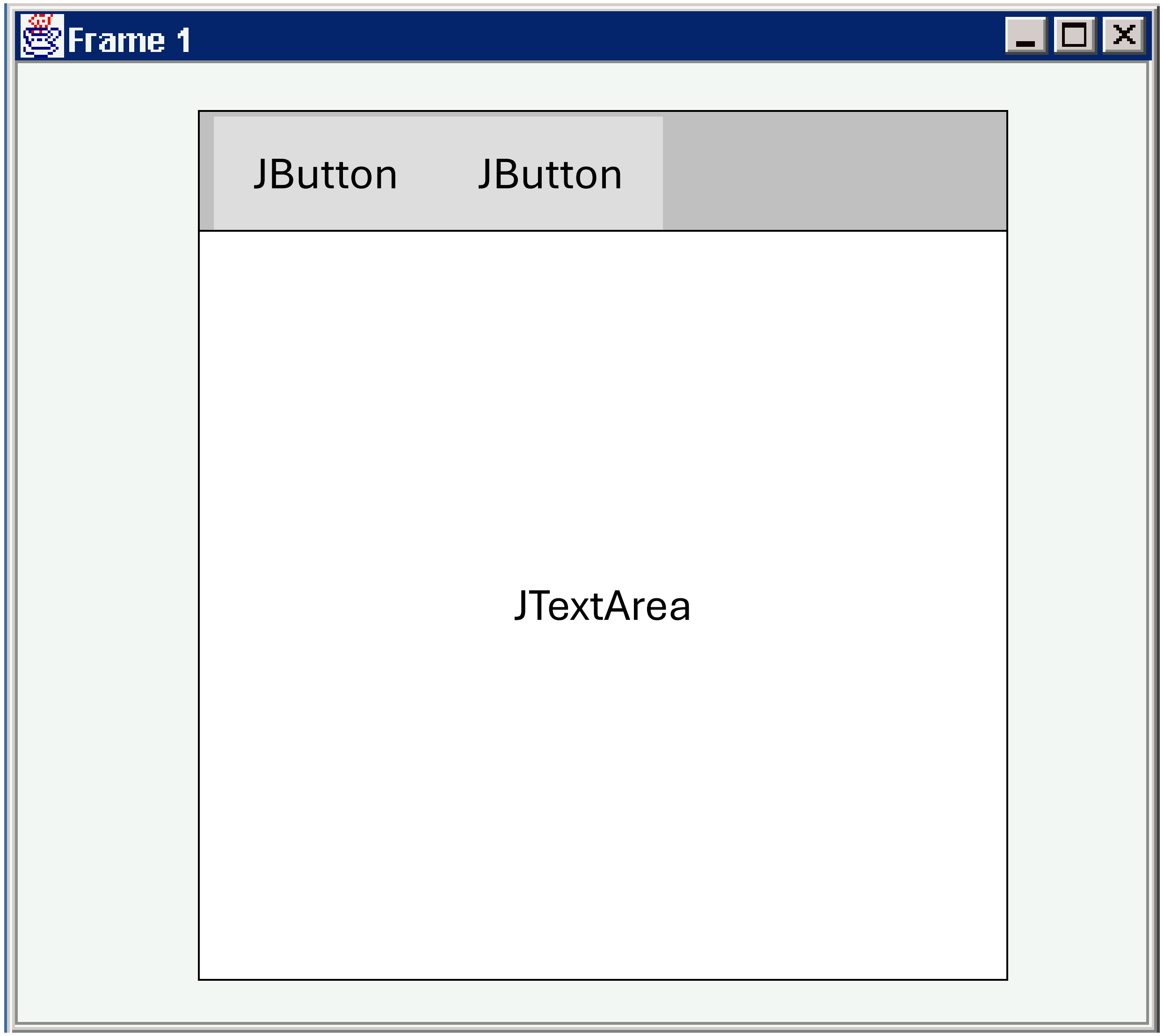


Grid Layout

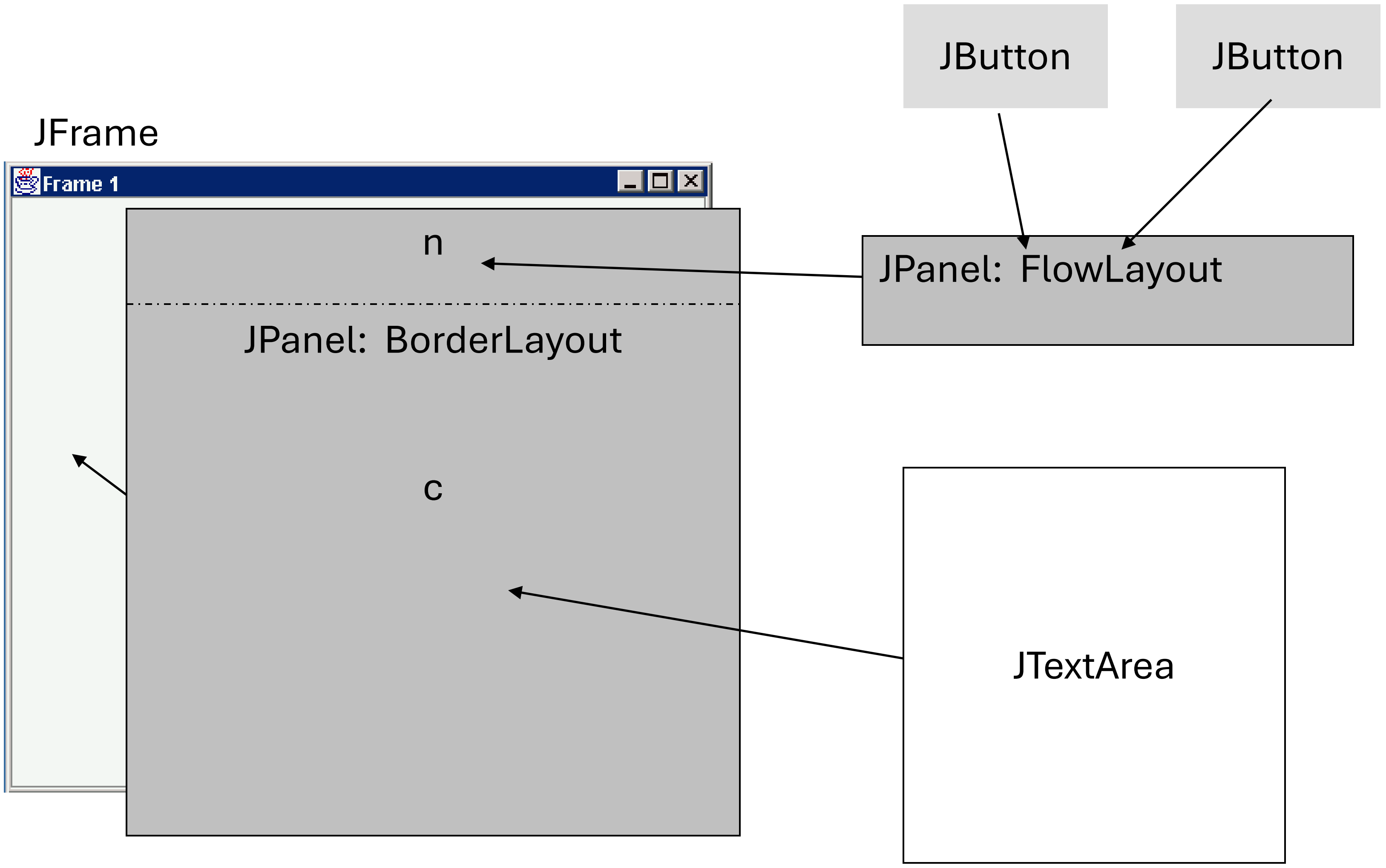
```
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.Container;
import java.awt.GridLayout;
import java.awt.Label;
class SimpleGUI3 extends JFrame{
    public SimpleGUI3()
    {
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        //add button
        JButton but1 = new JButton("Button 1");
        JButton but2 = new JButton("Button 2");
        Container cp = getContentPane();//must do this
        cp.setLayout(new GridLayout(2,2));
        cp.add(new Label("Label 1:"));
        cp.add(but1);
        cp.add(new Label("Label 2:"));
        cp.add(but2);

        setTitle("Grid Layout ");
        setVisible(true);
    }
     public static void main(String[] args)
     {
        SimpleGUI3 gui = new SimpleGUI3();
        gui.setSize(400,200);
     }   }
```

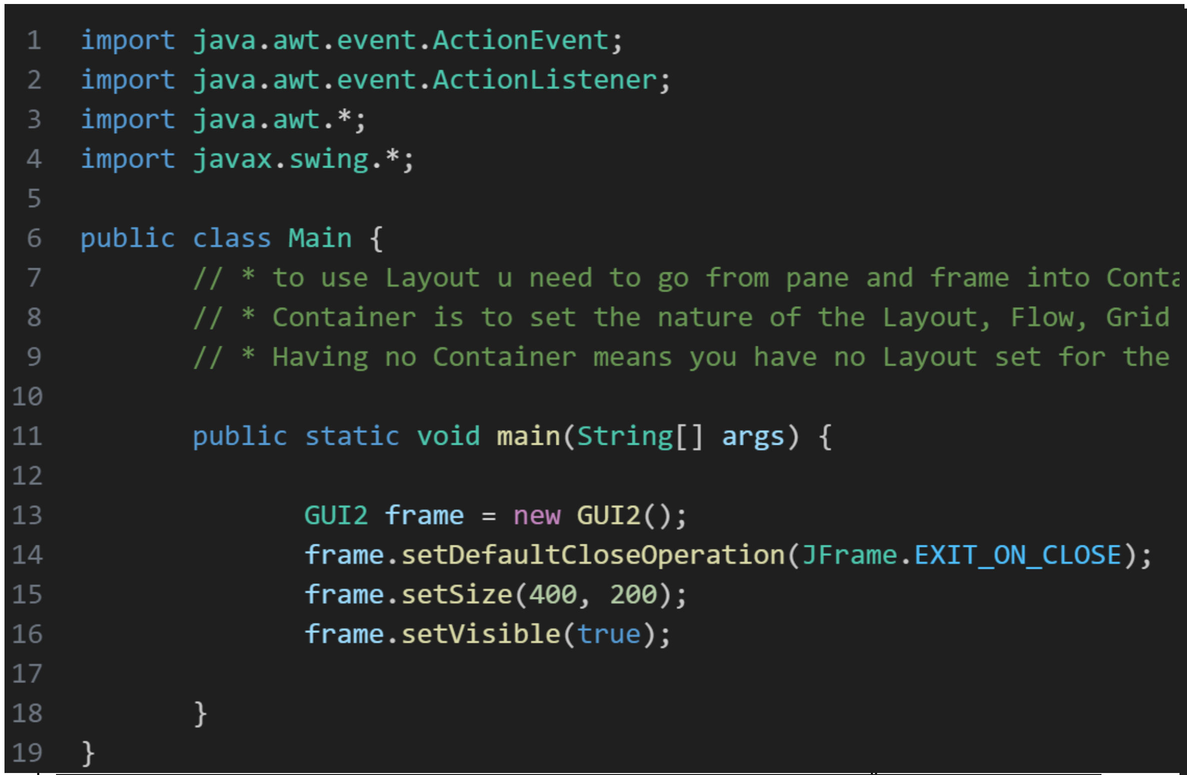


Combinations

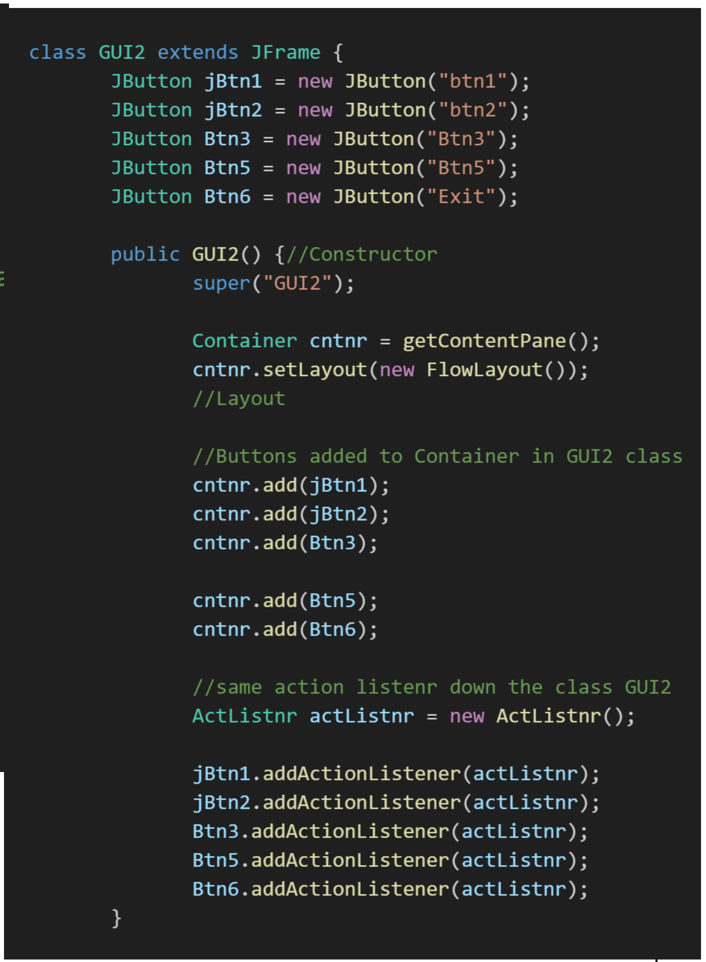


Combinations

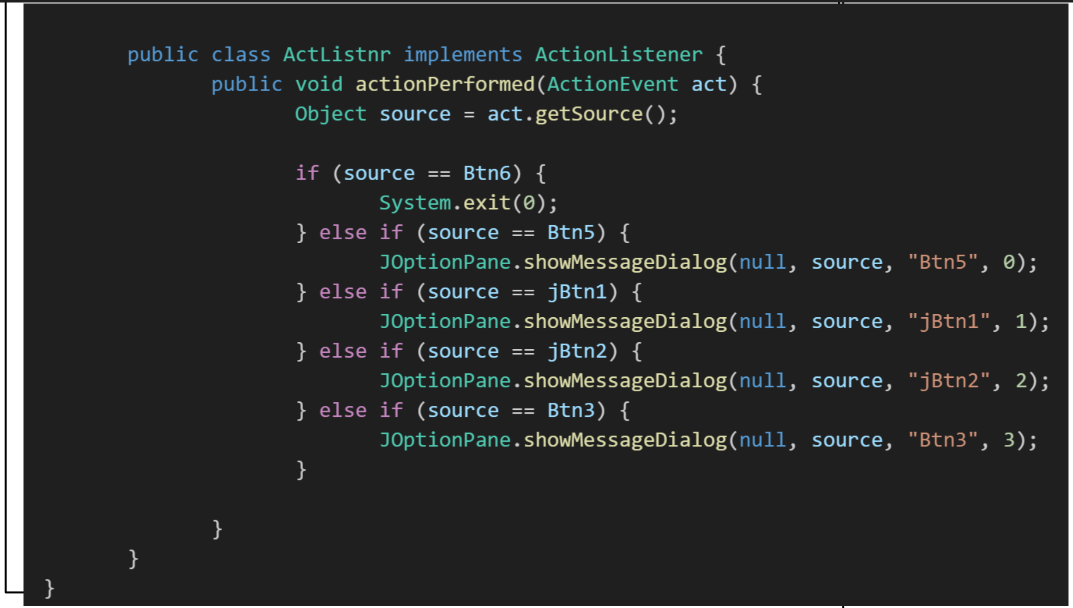




```java
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.*;
import javax.swing.*;

public class Main {
	// * to use Layout u need to go from pane and frame into Conta
	// * Container is to set the nature of the Layout, Flow, Grid
	// * Having no Container means you have no Layout set for the

	public static void main(String[] args) {

		GUI2 frame = new GUI2();
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setSize(400, 200);
		frame.setVisible(true);

	}
}
```

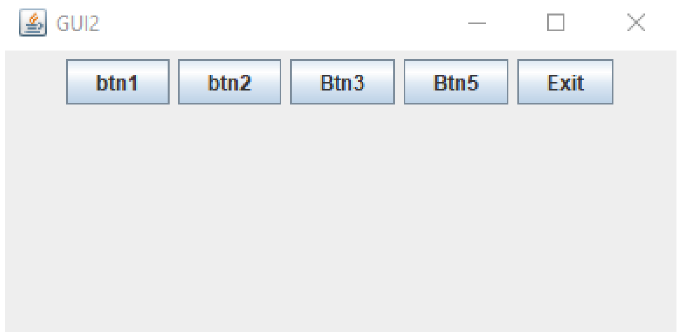


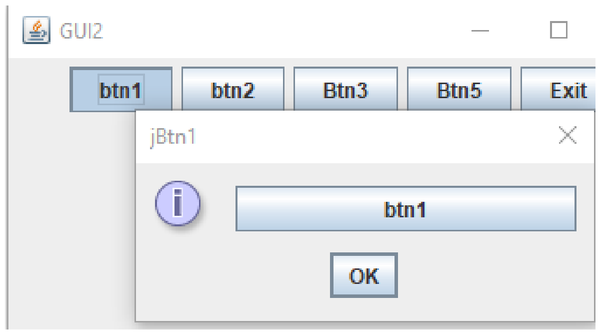
```java
class GUI2 extends JFrame {
	JButton jBtn1 = new JButton("btn1");
	JButton jBtn2 = new JButton("btn2");
	JButton Btn3 = new JButton("Btn3");
	JButton Btn5 = new JButton("Btn5");
	JButton Btn6 = new JButton("Exit");

	public GUI2() {//Constructor
		super("GUI2");

		Container cntnr = getContentPane();
		cntnr.setLayout(new FlowLayout());
		//Layout

		//Buttons added to Container in GUI2 class
		cntnr.add(jBtn1);
		cntnr.add(jBtn2);
		cntnr.add(Btn3);

		cntnr.add(Btn5);
		cntnr.add(Btn6);

		//same action listenr down the class GUI2
		ActListnr actListnr = new ActListnr();

		jBtn1.addActionListener(actListnr);
		jBtn2.addActionListener(actListnr);
		Btn3.addActionListener(actListnr);
		Btn5.addActionListener(actListnr);
		Btn6.addActionListener(actListnr);
	}
```

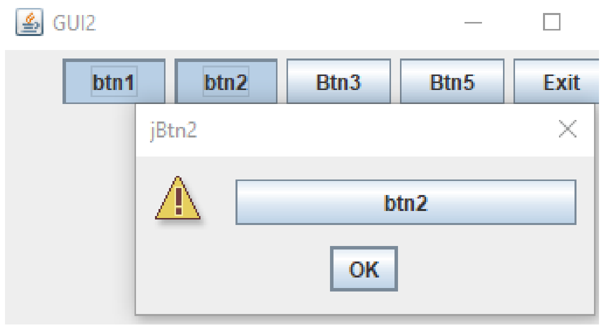


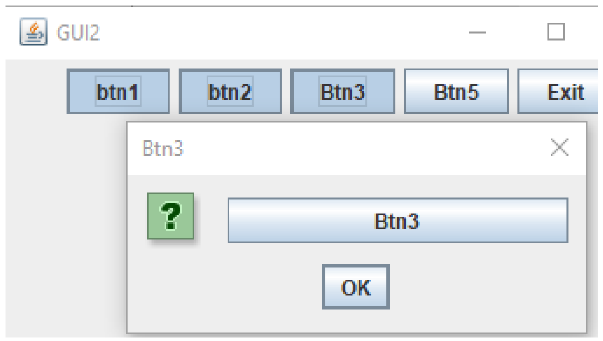
```java
	public class ActListnr implements ActionListener {
		public void actionPerformed(ActionEvent act) {
			Object source = act.getSource();

			if (source == Btn6) {
				System.exit(0);
			} else if (source == Btn5) {
				JOptionPane.showMessageDialog(null, source, "Btn5", 0);
			} else if (source == jBtn1) {
				JOptionPane.showMessageDialog(null, source, "jBtn1", 1);
			} else if (source == jBtn2) {
				JOptionPane.showMessageDialog(null, source, "jBtn2", 2);
			} else if (source == Btn3) {
				JOptionPane.showMessageDialog(null, source, "Btn3", 3);
			}

		}
	}
}
```

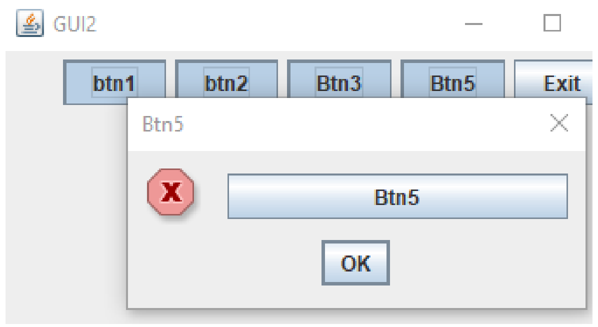
Same file, notice that the actionlistenr is part of the GUI2 JFrame


GUI2
btn1
btn2
Btn3
Btn5
Exit


GUI2
btn1
btn2
Btn3
Btn5
Exit
jBtn1
btn1
OK


GUI2
btn1
btn2
Btn3
Btn5
Exit
jBtn2
btn2
OK


GUI2
btn1
btn2
Btn3
Btn5
Exit
Btn3
Btn3
OK


GUI2
btn1
btn2
Btn3
Btn5
Exit
Btn5
Btn5
OK