GUI – Components, Layout and Appearance

2 – 4 pm Tuesday 8/12/2008 @JD2211

1

Agenda

- Review
  - Polymorphism
- GUI
  - java.awt and java.swing
  - Window (JFrame), JComponets (JPanel, JTextField, JButton)
  - Adding components to containers
  - Layout managers

2

Hello World – Using Dialog Boxes

```
import javax.swing.JOptionPane;

public class Main {
	public static void main(String[] args) {
		JOptionPane.showMessageDialog(null, "Hello World!");
		System.exit(0);
    }
}
```

3

Hello World – Using Dialog Boxes

```
import javax.swing.JOptionPane;

public class Main {
	public static void main(String[] args) {
		String name;
		String message;

		name = JOptionPane.showInputDialog("What is your name");
		message = "Hello " + name;
		JOptionPane.showMessageDialog(null, message);
		System.exit(0);
	}
}
```

4

showMessageDialog

- `JOptionPane.showMessageDialog(null, "Hello World!");`
- `JOptionPane.showMessageDialog (null, message, "Hello Message", JOptionPane.INFORMATION_MESSAGE);`

5

Message Type

- ERROR_MESSAGE 
- INFORMATION_MESSAGE 
- WARNING_MESSAGE 
- QUESTION_MESSAGE 
- PLAIN_MESSAGE

6

Java Standard Class Library

Packages	Provides support to
java.applet	Create programs (applets)
java.awt	Draw graphics and create GUI; AWT stands for Abstract Window Tool kit
java.beans	Define software components that can be easily combined into applications
java.io	Perform a variety of input and output functions
java.lang	General support; it is automatically imported into all Java programs
java.math	Perform calculations
java.net	Communicate across a network
java.rmi	Create programs that can be distributed across among computers; RMI stands for Remote Method Invocation
java.security	Enforce security restrictions
java.sql	Interact with databases
java.text	Format text for output
java.util	General utilities
java.swing	Creates GUI with components that extended the AWT applications
java.xml.parsers	Process XML document

7

GUI Components

- Java contains two sets of standard GUI classes AWT and Swing, which are defined in packages `java.awt` and `java.swing` respectively.

8

Comparison between AWT and Swing

- http://dn.codegear.com/article/26970
- In general, AWT components are appropriate for simple applet development or development that targets a specific platform (i.e. the Java program will run on only one platform). For most any other Java GUI development you will want to use Swing components.

9

AWT Components

- Figure 6.1 (p. 157)

10

Swing

- `JApplet`
- `JDialog`
- `JFrame`
- `JWindow`
- `JComponent`
  - Figure 6.2 (p.158)

11

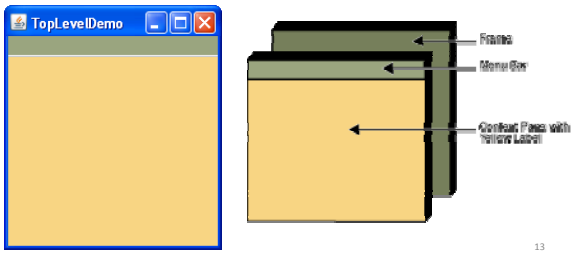
`JComponent`

- Class `JComponent` is the superclass of most Swing components. The class declares the common attributes and behaviors of all subclasses of `JComponents`, including
  - A look-and-feel
  - Shortcut keys
  - Common event-handling capabilities
  - Support for assistive technologies
  - Support for user interface localization

12

JFrame

- A JFrame is a window with a title bar and a border



13

Create a JFrame

```
import javax.swing.*;

public class Frame {

  public static void main(String s[]) {
    JFrame frame = new JFrame("JFrame Source Demo");
    frame.setSize(200,200);
    frame.setVisible(true);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}
```

14

Create a JFrame (cont'd)

```
import javax.swing.*;

public class Frame2 extends JFrame {
  public Frame2() {
    super("JFrame Demo");
    setSize(200,200);
    setVisible(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }

  public static void main(String[] args){
    Frame2 f2 = new Frame2();
  }
}
```

15

Adding `JComponents` **to** `Container`

- In applications with windows classes (`JFrame, JWindow, JApplet, JDialog`, and `JInternalFrame`), we add JComponents to the **`content pane`**, which is an object of class `Container`.

16

Adding `JComponents` **to** `Container`

- Content pane is called the window's "working area"
- There are two methods that originates in class Container
  - `add()`
  - `setLayout()`

17

`add()`

- Attach components to content pane

```
// a window w of type JFrame and
// want to position a Jbutton object b

w.getContentpane().add(b);
```

18

add() (cont'd)

- If we want to place several components in a window, it becomes clumsy to write `w.getContentpane()` in each location

```
Container c = w.getContentPane();
c.add(b);
```

19

setLayout()

- This method enables a program to specify the **`layout manager`** that helps a `Container` position and size its components

20

Layout Managers

- Layout managers are provided to arrange GUI components in a container for presentation purposes.
- The layout managers provide basic layout capabilities that are easier to use than determining the exact position and size of every GUI components

21

Layout Managers (cont'd)

Layout Mangers	Descriptions
FlowLayout	Default for java.awt.Applet, java.awt.Panel, and javax.swing.JPanel. Places components sequentially (left to right) in the order they were added. It is also possible to specify the order of the components by using the Container method add, which takes a Component and an integer index position as arguments.
BorderLayout	Default for the content panes of JFrames (ant other windows) and JApplets. Arranges the components into five areas: NORTH, SOUTH, EAST, WEST and CENTER
GridLayout	Arranges the components into rows and columns.

```
c.setLayout(new FlowLayout());
// class FlowLayout inherits from class Object and
//implementations interface LayoutManager, which declares
//the methods a layout manager uses to arrange and size
//GUI components in a container
```

22

Size of Components

- The size of a window can be specified in two ways
  - `setSize(int width, int height)`
  - `Pack()`

23

Size of Components (cont'd)

- If you want an internal component to be a certain size, call the methods `setMinimumSize`, `setMaximumSize`, and `setPreferredSize`

```
c.setMinimumSize(new Dimension(100, 50);
```

24

Size of Components (cont'd)

- The `LayoutManager` does not calculate the size of internal components directly. This is not done until the `LayoutManager` needs to decide how the components are to be placed in the container, which is not normally done until the surrounding window is made visible or you call `pack`

25

Drawing of the Components

- All components that belongs to class `JComponent` or one of its subclasses have a pre-defined method called `paintComponent`, which is called automatically each time the component needs to be redrawn.
- Note that your program should never call method `paintComponent`. This is done automatically.

26

Java GUI

- To use GUI components effectively, the `javax.swing` and `java.awt` inheritance hierarchies must be understood – especially class `Component`, class `Container` and class `JComponent`, which declare features common to most Swing component

27

Lab Assignment

- http://www.csun.edu/~twang/Java/Lab/8-12-Lab.pdf

28