Graphical User Interfaces

CSC 1051 – Data Structures and Algorithms I

Dr. Mary-Angela Papalaskari
Department of Computing Sciences
Villanova University

Course website:
www.csc.villanova.edu/~map/1051/
Graphical User Interface (GUI)

• A GUI component
  – an object that represents a screen element
  – examples: buttons, text fields, labels, panels, frames

• GUI-related classes from packages:
  – java.awt Abstract Windowing Toolkit (AWT): the original Java GUI package
  – javax.swing Swing provides additional and more versatile components
  – Both packages are needed to create a Java GUI-based program
GUI Container

A kind of component:

• hold and organize other components
• two main types:

**frame**: a container displayed as a separate window

• It can be repositioned and resized on the screen as needed
• has its own title bar with close-minimize-resize buttons
• a *heavyweight* container: managed by the underlying operating system

**panel** – a container that cannot be displayed on its own but is used to organize other components

• A panel must be added to another container (like a frame or another panel) to be displayed
• a *lightweight* container: managed by the Java program itself
Other components: Labels

• A *label* is a GUI component that displays a line of text and/or an image

• Let's look at a program that organizes two labels in a panel and displays that panel in a frame

• This program is not interactive, but the frame can be repositioned and resized

• See `Authority.java`
import java.awt.*;
import javax.swing.*;

public class Authority
{
   // Displays some words of wisdom.
   public static void main (String[] args)
   {
      JFrame frame = new JFrame ("Authority");

      frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

      JPanel primary = new JPanel();
      primary.setBackground (Color.yellow);
      primary.setPreferredSize (new Dimension(250, 75));

      continued
continued

    JLabel label1 = new JLabel("Question authority,");
    JLabel label2 = new JLabel("but raise your hand first.");

    primary.add(label1);
    primary.add(label2);

    frame.getContentPane().add(primary);
    frame.pack();
    frame.setVisible(true);
}
continued

JLabel label1 = new JLabel("Question authority,");
JLabel label2 = new JLabel("but raise your hand first.");

primary.add(label1);
primary.add(label2);

frame.getContentPane().add(primary);
frame.pack();
frame.setVisible(true);
Nested Panels

• Containers that contain other components make up the *containment hierarchy* of an interface

• This hierarchy can be as intricate as needed to create the visual effect desired

• The following example nests two panels inside a third panel – note the effect this has as the frame is resized

• See [NestedPanels.java]( NestedPanels.java)
import java.awt.*;
import javax.swing.*;

public class NestedPanels
{
    // Presents two colored panels nested within a third.
    public static void main (String[] args)
    {
        JFrame frame = new JFrame ("Nested Panels");
        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

        // Set up first subpanel
        JPanel subPanel1 = new JPanel();
        subPanel1.setPreferredSize (new Dimension(150, 100));
        subPanel1.setBackground (Color.green);
        JLabel label1 = new JLabel ("One");
        subPanel1.add (label1);
continued

// Set up second subpanel
JPanel subPanel2 = new JPanel();
subPanel2.setPreferredSize (new Dimension(150, 100));
subPanel2.setBackground (Color.red);
JLabel label2 = new JLabel ("Two");
subPanel2.add (label2);

// Set up primary panel
JPanel primary = new JPanel();
primary.setBackground (Color.blue);
primary.add (subPanel1);
primary.add (subPanel2);

frame.getContentPane().add(primary);
frame.pack();
frame.setVisible(true);
// Set up second subpanel
JPanel subPanel2 = new JPanel();
subPanel2.setPreferredSize(new Dimension(150, 100));
subPanel2.setBackground(Color.red);
JLabel label2 = new JLabel("Two");
subPanel2.add(label2);

// Set up primary panel
JPanel primary = new JPanel();
primary.setBackground(Color.blue);
primary.add(subPanel1);
primary.add(subPanel2);
frame.getContentPane().add(primary);
frame.pack();
frame.setVisible(true);
Images

• Images can be displayed in a Java program in various ways

• As we've seen, a JLabel object can be used to display a line of text

• It can also be used to display an image

• That is, a label can be composed of text, an image, or both at the same time
Images

• The ImageIcon class is used to represent the image that is stored in a label

• If text is also included, the position of the text relative to the image can be set explicitly

• The alignment of the text and image within the label can be set as well

• See LabelDemo.java
import java.awt.*;
import javax.swing.*;

public class LabelDemo {
    public static void main (String[] args) {
        JFrame frame = new JFrame("Label Demo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        ImageIcon icon = new ImageIcon("devil.gif");

        JLabel label1, label2, label3;

        label1 = new JLabel("Devil Left", icon, SwingConstants.CENTER);
continued
continued

label2 = new JLabel ("Devil Right", icon, SwingConstants.CENTER);
label2.setHorizontalTextPosition (SwingConstants.LEFT);
label2.setVerticalTextPosition (SwingConstants.BOTTOM);

label3 = new JLabel ("Devil Above", icon, SwingConstants.CENTER);
label3.setHorizontalTextPosition (SwingConstants.CENTER);
label3.setVerticalTextPosition (SwingConstants.BOTTOM);

JPanel panel = new JPanel();
panel.setBackground (Color.cyan);
panel.setPreferredSize (new Dimension (200, 250));
panel.add (label1);
panel.add (label2);
panel.add (label3);

frame.getContentPane().add(panel);
frame.pack();
frame.setVisible(true);
}
}
label2 = new JLabel("Devil Right", icon, SwingConstants.CENTER);
label2.setHorizontalTextPosition(SwingConstants.LEFT);
label2.setVerticalTextPosition(SwingConstants.BOTTOM);

label3 = new JLabel("Devil Above", icon, SwingConstants.CENTER);
label3.setHorizontalTextPosition(SwingConstants.CENTER);
label3.setVerticalTextPosition(SwingConstants.BOTTOM);

JPanel panel = new JPanel();
panel.setBackground(Color.cyan);
panel.setPreferredSize(new Dimension(200, 250));
panel.add(label1);
panel.add(label2);
panel.add(label3);
frame.getContentPane().add(panel);
frame.pack();
frame.setVisible(true);
Homework

• **Reading**: Sections 3.9-3.11
• Review your work for **Lab 7**