

# Advanced GUI

---

Introduction to Java  
CIS 226  
Brad Rippe

---

---


---

---

---

---

---



# Agenda

- Advanced GUI
  - More Swing Components
  - JTextArea
  - JPanel
  - JSlider
  - Applets/Applications
  - Menus
  - JPopupMenu
  - Pluggable Look and Feel
  - JDesktopPane and JInternalFrame
  - More Layout Managers

---

---


---

---

---

---

---



# JTextArea

- Swing's text components display text and optionally allow the user to edit the text.
- Use a text area to allow the user to enter unformatted text of any length or to display unformatted help information.
- Doesn't have scroll bars by default

---

---

---

---

---

---

---

## JPanels

- **JPanel**

- Does not support conventional events
  - e.g., events offered by buttons, text areas, etc.
- Capable of recognizing lower-level events
  - e.g., mouse events, key events, etc.
- Self-contained panel
  - Listens for its own mouse events

---

---

---

---

---

---

---

---

## SelfContainedPanel Example

---

---

---

---

---

---

---

---

## JSlider

- let the user enter a numeric value bounded by a minimum and maximum value.
- By using a slider instead of a text field, you eliminate input errors.
  - Enable users to select from range of integer values
  - Several features
    - Tick marks (major and minor)
    - Snap-to ticks
    - Block marks
    - Orientation (horizontal and vertical)

---

---

---

---

---

---

---

---

## JSlider

- By default, spacing for major and minor tick marks is zero.
- To see tick marks, you must explicitly set the spacing for either major or minor tick marks (or both) to a non-zero value and call `setPaintTicks(true)`.
- Just calling `setPaintTicks(true)` is not enough.
- To display standard, numeric labels at major tick mark locations, set the major tick spacing, then call `setPaintLabels(true)`.

---

---

---

---

---

---

---

---

## JSlider

- When you move the slider's knob, the `stateChanged` method of the slider's `ChangeListener` is called.
- Here is the code for this example's change listener:

```
class SliderListener implements ChangeListener {
    public void stateChanged(ChangeEvent e) {
        JSlider source = (JSlider)e.getSource();
        if (!source.getValueIsAdjusting()) {
            int fps = (int)source.getValue();
            if (fps == 0) {
                if (!frozen)
                    stopAnimation();
            } else {
                delay = 1000 / fps;
                timer.setDelay(delay);
                timer.setInitialDelay(delay * 10);
                if (frozen) startAnimation();
            }
        }
    }
}
```

• Notice that our `stateChanged` method changes the animation speed only if `getValueIsAdjusting` returns false. Many change events are fired as the user moves the slider knob. This program is interested only in the final result of the user's

---

---

---

---

---

---

---

---

## Timer

- class fires one or more action events after a specified delay.
- To perform a task once, after a delay.  
For example, the tool tip manager uses timers to determine when to show a tool tip and when to hide it.
- To perform a task repeatedly.  
For example, you might perform animation or update a component that displays progress toward a goal. See [Creating an Animation Loop with Timer](#) for an example and discussion of using a timer for animation.

---

---

---

---

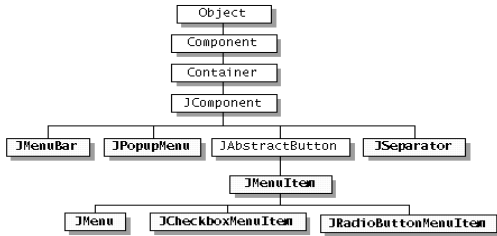
---

---

---

---

## Menu



---

---

---

---

---

---

---

---

## Menu

- JMenuBar – the container for menus. Added to a JFrame with setJMenuBar() method
- JMenu – Menus contain menu items and are added to menu bars or to other menus as submenus. When a menu is clicked, the menu expands to show its list of menu items. Clicking a menu item generates an action event.
- JMenuItem – used to initiate an action or it can be a submenu that provides more menu items from which the user can select. Submenus are useful for grouping related menu items in a menu.

---

---

---

---

---

---

---

---

## PopupMenu

- Cosmetology App Example

---

---

---

---

---

---

---

---

## Pluggable Look and Feel

- Java awt programs allow programs running on their target platforms to look like other applications on that target platform.
- Swing allows the programmer to change the look and feel of applications

---

---

---

---

---

---

---

---

## Pluggable Look and Feel

Here are some of the arguments you can use for setLookAndFeel:

- UIManager.getCrossPlatformLookAndFeelClassName ()
  - Returns the string for the one look-and-feel guaranteed to work -- the Java Look & Feel.
- UIManager.getSystemLookAndFeelClassName ()
  - Specifies the look and feel for the current platform. On Win32 platforms, this specifies the Windows Look & Feel. On Mac OS platforms, this specifies the Mac OS Look & Feel. On Sun platforms, it specifies the CDE/Motif Look & Feel.
- "javax.swing.plaf.metal.MetalLookAndFeel"
  - Specifies the Java Look & Feel. (The codename for this look and feel was *Meta!*) This string is the value returned by the getCrossPlatformLookAndFeelClassName method.
- "com.sun.java.swing.plaf.windows.WindowsLookAndFeel"
  - Specifies the Windows Look & Feel. Currently, you can use this look and feel only on Win32 systems.
- "com.sun.java.swing.plaf.motif.MotifLookAndFeel"
  - Specifies the CDE/Motif Look & Feel. This look and feel can be used on any platform.
- "javax.swing.plaf.mac.MacLookAndFeel"
  - Specifies the Mac OS Look & Feel, which can be used only on Mac OS platforms.

---

---

---

---

---

---

---

---

## Pluggable Look and Feel

```
try {  
    UIManager.setLookAndFeel("com.sun.java.swing.plaf.  
        windows.WindowsLookAndFeel");  
    SwingUtilities.updateComponentTreeUI(this);  
} catch (Exception e) {  
    //e.printStackTrace();  
    logger.log("Look & Feel Error", e.getMessage());  
}  
this.setSize(Toolkit.getDefaultToolkit().getScreenSize());
```

---

---

---

---

---

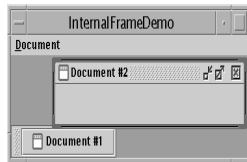
---

---

---

## JDesktop and JInternalFrame

- With the `JInternalFrame` class, you can display a `JFrame`-like window within another window.
- add internal frames to a desktop pane.
- The desktop pane, in turn, might be used as the content pane of a `JFrame`.
- (MDI) Multiple Document Interface



---

---

---

---

---

---

---

---

## To do:

- Files & Streams
- JSP & Servlets – Tomcat

---

---

---

---

---

---

---

---