JAVA Graphical User Interface
JAVA Swing
Introduction

• Swing – A set of GUI classes
  – Part of the Java's standard library
  – Much better than the previous library: AWT
    • Abstract Window Toolkit
  – Contents and shape are separated (MVC support)
  – Fine-grained control over the behavior and look and feel
  – Platform independent
    • Isolates the programmer from the operating system's GUI
Swing component inheritance hierarchy

- **Component** defines methods used in its subclasses
  - (for example, `paint` and `repaint`)
- **Container** - collection of related components
  - When using `JFrame`, add components to content pane (a `Container`)
- **JComponent** - superclass to most Swing components
Swing components

• Containers
  – Contain and manage other components.
  – Top Level/Internal
  – Examples: JFrame (Top Level), JScrollPane, JPanel.

• Basic controls
  – Atomic components
  – Used for showing output and/or getting some input
  – Inherits JComponent
  – Examples: JButton, JLabel, JTextArea, JTable, JList

• Every Swing class extends the corresponding AWT class
  – For backward-compatibility reasons
Jcomponent features

- Pluggable look and feel
  - Can look like different platforms, at run-time

- Shortcut keys (mnemonics)
  - Direct access to components through keyboard

- Tool tips
  - Describe component when mouse rolls over it
JComponent Methods

• `setVisible` (boolean mode) true means visible

• `setToolTipText` (String toolTip) mouse hover shows this text

• `setForeground` (Color foreColor) foreground color of the component

• `setBackground` (Color backcolor) background color of the component

• `setOpaque` (boolean mode) background is opaque if true and transparent if false

• `setEnabled` (boolean mode) enabled if true and can respond to user input
Jframe features

• Constructors:
  – JFrame() no text in the title bar
  – JFrame (String titlebartext)

• Methods:
  – getContentPane() – returns the content pane object for the window
  – setDefaultCloseOperation( int operation) sets the default operation when the user closes the “X” in the upper corner
  – setSize (int width, int height) sets the window size in pixels
  – setVisible (boolean mode) displays window if mode = true; hides if false
GUI application tasks

• Call constructor of the JFrame superclass

• Get an object reference to the content pane container - GUI objects are added to this pane

• Set the layout manager to arrange GUI components in the window

• Instantiate each component

• Add each component to the content pane

• Set the window size

• Display the window
import javax.swing.*;
import java.awt.BorderLayout;

public class First {
    public static void main(String[] args) {
        JFrame frame = new JFrame("My First Frame");
        // operation to do when the window is closed.
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(new BorderLayout());
        frame.getContentPane().add(new JLabel("I Love Swing"), BorderLayout.CENTER);
        frame.pack();
        frame.setVisible(true);
    }
}
JDialog

- JDialog is directly descended from the Dialog class
- JDialog contains a rootPane hierarchy including a contentPane
- All dialogs are modal, which means the current thread is blocked until user interaction with it has been completed.
- JOptionPane.showMessageDialog – reports something that has happened to the user
- JOptionPane.showConfirmDialog – asks for a confirming response e.g. yes, no, cancel
- JOptionPane.showInputDialog – prompt the user for input
**JFileChooser**

- `javax.swing.JFileChooser`:
  - Allows the user to choose a file
  - Supports “open” and “save”:
    - `showOpenDialog()`, `showSaveDialog()`

- **Example**

  ```java
  JFileChooser fc = new JFileChooser();
  int returnVal = fc.showOpenDialog(null);
  if(returnVal == JFileChooser.APPROVE_OPTION)
      System.out.println("File: " + fc.getSelectedFile());
  ```
Layout

- Each container has a layout manager
  - Determines the size, location of contained widgets.

- Setting the current layout of a container:
  
  ```java
  void setLayout(LayoutManager lm)
  ```

- `LayoutManager implementing classes`:
  - BorderLayout
  - BoxLayout
  - FlowLayout
  - GridLayout
Layouts

• BorderLayout

Position must be specified, e.g. add ("North", myComponent)
Layouts

- **BoxLayout**
  - The BoxLayout class puts components in a single row or column.
  - It respects the components’ requested maximum sizes.
Layouts

• FlowLayout
  • FlowLayout is the default layout manager for every JPanel.
  • It simply lays out components from left to right, starting new rows if necessary
## Layouts

- **GridLayout**

  GridLayout simply makes a bunch of components equal in size and displays them in the requested number of rows and columns.
Menus

Menu Bar
- JMenuBar()
- add( JMenu )

Menu
- JMenu( String )
- add( JMenuItem )

JMenuItem( String )
JMenuItem( String,int )
JLabel

• Labels
  – Provide text instructions on a GUI
  – Read-only text
  – Programs rarely change a label's contents
  – Class JLabel (subclass of JComponent)

• Methods
  – Can declare label text in constructor
  – myLabel.setToolTipText( "Text" )
  – myLabel.setText( "Text" )
  – myLabel.getText()
JButton

- **Methods of class JButton**
  - **Constructors**
    
    ```java
    JButton myButton = new JButton("Label");
    JButton myButton = new JButton("Label", myIcon);
    ```
  - **setRolloverIcon** (myIcon)
    - Sets image to display when mouse over button
  - **Class** `ActionEvent` `getActionCommand`
    - returns label of button that generated event
JCheckBox

- **When JCheckBox changes**
  - ItemEvent generated
    - Handled by an ItemListener, which must define itemStateChanged
    - Register handlers with with addItemListener

- **Class ItemEvent**
  - getStateChange
    - Returns ItemEvent.SELECTED or ItemEvent.DESELECTED
JRadioButton

• Radio buttons
  – Have two states: selected and deselected
  – Normally appear as a group
    • Only one radio button in group selected at time
    • Selecting one button forces the other buttons off
  – Mutually exclusive options
  – ButtonGroup - maintains logical relationship between radio buttons

• Class JRadioButton
  – Constructor
    • JRadioButton( "Label", selected )
    • If selected true, JRadioButton initially selected
JList

• List
  – Displays series of items
  – may select one or more items

• Class JList
  – Constructor JList( arrayOfNames )
    • Takes array of Objects (Strings) to display in list
  – setVisibleRowCount( n )
    • Displays n items at a time
    • Does not provide automatic scrolling
More about Swing on

http://java.sun.com/docs/books/tutorial/uiswing
END