Swing

A Quick Tutorial on Programming Swing Applications
MVC – Model View Controller

- Swing is based on this design pattern
- It means separating the implementation of an application into layers or components:
  - The **Model** - the data structure that represents something (like a customer info rec)
  - The **Controller** - the user interface logic for manipulating it
  - The **View** - the display of that data structure to the user.
What is Swing?

• A set of classes (part of JFC) that support platform independent GUI (Graphical User Interface)

• Successor to the original Java GUI classes (AWT) which didn’t work very well (they had platform dependencies that really made it a difficult API to use)

• AWT wasn’t very “sexy”
Swing

• Visible “widgets” - windows, buttons, combo boxes, trees, tables, checkboxes, text fields, menus, …
• Containers of components – applets, dialogs, windows and frames
• Supporting classes and utility methods
Some important Swing visible component classes

- JApplet **
- JButton
- JCheckBox
- JColorChooser
- JComboBox
- JDialog **
- JFileChooser
- JFormattedTextField
- JFrame **
- JLabel
- JList
- JMenu
- JMenuBar
- JMenuItem
- JPanel
- JPasswordField
- JPopupMenu
- JProgressBar
- JRadioButton
- JScrollPane
- JSlider
- JSpinner
- JTable
- JTextArea
- JTextField
- JToggleButton
- JToolBar
- JTree
- JWindow **
- ** means a top level containers
Using netbeans to create a JDialog
adding fields
Steps
1. Choose Template
2. New Object Name
3. Basic Class Definition
4. Create Fields
5. Override Inherited Methods
6. Create Methods

Create Methods
Add a method to the Methods list, then fill it in below. Click an item's "..." button for its customization.

Methods:
- initComponents()
- closeDialog(java.awt.event.WindowEvent)
- main(String[])
- getCustomerInfoFromFileName(String)
- saveCustomerInfoToFile()

Name:
saveCustomerInfoToFile

Return Type:
boolean

Parameters (comma-separated):
String fileNameToSaveTo

Exceptions (comma-separated):

Access Level:
<default>

Method Type:
- <default>
  - final
  - static
  - native
  - synchronized
  - abstract
my empty
CustomerInfoDialog:JDialog
code created
To kill a zombie or running process in netbeans right click and choose: ”terminate”
executing the class displays:
Editing a dialog

• 1\textsuperscript{st} select a layout manager for the dialog
changing the layout manager
what layout manager should I use?

• Start with the absolute and then experiment when you feel comfortable (or hire a graphic artist and let them worry about it ;-).
Adding other components to the view - JTextFields
execute the class
Adding a combo box
edit the model property for the combo box

type in state abbreviations separated by commas
preferred size property
MVC

Model – View – Controller Design Pattern
Design Patterns

• A design pattern is a way of designing code that benefits from experience of other developers – see GoF (Gang of Four) on Patterns
• Design patterns are “rules of thumb” & best practices
• A GUI is based on many design patterns
  – 3D Pliancy
  – Feedback
  – Icons
  – Menus
  – Pointing
  – Mnemonics & Accelerators
  – Many more …
• A pattern usually has a name (and several aliases), a context, a problem it addresses, a description of the solution, hints of when to use it and when not to.
MVC – Model View Controller pattern

- Swing components are designed as MVC components
  - **Model** = data or object that is the to be visually represented
  - **View** = one or more visual representations of that data/object
  - **Controller** = code to manage input to the model
MVC

- © Sun 2002
MVC in Swing Components

- The Swing component class is the view and controller
- A separate class is the model
- Most components come with a default model
- You can set the model to your own model for a control
- Several controls could share a model!
Creating icons using the blank icon to start with
Change the properties of the button to use your icon.
Pushing the buttons changes the displayed prices.
A different and better layout

The NYU Cafe' UI is a virtual simulated Cafe'. Its purpose is to provide a tangible business problem to study in the UI class.

Hours: 24 x 7
(its virtual and fully automated)
private javax.swing.JPasswordField jPasswordField1;
private javax.swing.JTree jTree1;
private javax.swing.JSlider jSlider1;
private javax.swing.JProgressBar jProgressBar1;
private javax.swing.JTable jTable1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton1;
private javax.swing.JFileChooser jFileChooser1;
private javax.swing.JLabel jLabel1;
Swing based MenuLookDemo
TopLevelWindows.java
package SwingSamples;
import javax.swing.*;
public class TopLevelWindows
{
    public static void main(String args[])
    {
        JFrame myJFrame = new JFrame("The JFrame");
        myJFrame.setSize(300,300);
        myJFrame.setLocation(100,100);

        JWindow myJWindow = new JWindow();
        myJWindow.setSize(300,300);
        myJWindow.setLocation(500, 100);

        myJFrame.setVisible(true);
        myJWindow.setVisible(true);
    }
}
Top Level Containers

- Must have a top level container in Swing
- You must add components to the associated content pane
package SwingSamples;

import java.awt.*;
import javax.swing.*;

public class ContentPaneExample {

    public static void main(String args[]) {
        JFrame myJFrame = new JFrame("JFrame");
        myJFrame.setLocation(100, 100);

        Container myContentPane = myJFrame.getContentPane();
        myContentPane.setLayout(new FlowLayout());
        myContentPane.add(new JLabel("One"));
        myContentPane.add(new JLabel("Two"));

        myJFrame.pack(); // reformats the layout to the minimum size to fit everything
        myJFrame.setVisible(true);
    }
}

Poelman & Associates, Inc. (c) 2003
ContentPaneExample2.java

package SwingSamples;

import java.awt.*;
import javax.swing.*;

public class ContentPaneExample2
{
    public static void main(String args[])
    {
        JFrame myJFrame = new JFrame("JFrame");
        myJFrame.setLocation(100,100);

        Container myContentPane = new JPanel();
        myContentPane.add(new JLabel("One"));
        myContentPane.add(new JLabel("Two"));

        myJFrame.setContentPane(myContentPane);
        myJFrame.pack();
        myJFrame.setVisible(true);
    }
}
Events

- Swing uses them to communicate between swing components.
- An event is just a method call on the receiving object by the sending object. The method passes the event object.
  ```java
  addActionListener(ActionListener listener);
  removeActionListener(ActionListener listener);
  ```
- An object registers to receive events. The method that gets called is:
  ```java
  actionPerformed(ActionEvent e);
  ```
Events

• In Swing they are multicast – 1 to many possible. Manes multiple method calls by the send basically.
• Order isn’t defined, though.
• Events are immutable to the receiver.
• Events may be queued as in the keyboard event queue.
• Multiple events maybe compressed into one as in mouse movements.
Event Modifier Flags

- SHIFT_MASK
- CTRL_MASK
- META_MASK
- ALT_MASK
- BUTTON1_MASK
- BUTTON2_MASK
- BUTTON3_MASK
- Detect when certain keys are also pressed.

```java
int modifierFlags = myEvent.getModifiers();
if ((modifierFlags & InputEvent.CRTL_MASK)!=0)
    System.println.out("Pressing the contrl key");
```
Event Types

- ComponentEvent  //resized, moved, shown, hidden
- FocusEvent      //gained, lost
- KeyEvent        //typed, pressed, released
- MouseEvent      //clicked, pressed, released, entered, exited
- ContainerEvent  //componentAdded  componentRemoved
- ActionEvent     //fired by: JButton, JCheckBox, ...
- AdjustmentEvent //fired by: JScrollBar

- Many more ....
Event Adapter Classes

• Map incoming events to a method to invoke on the model to achieve the function.
• Separates the View & Controller from the Model (MVC)
• Prebuilt adapter has stubbed out methods for events. You only implement the ones you are interested. You do this by extending the adapter and overriding the methods you need.
• Follows a general design pattern of called “adapter”.
• MouseAdapter, MouseInputAdapter, MouseMotionAdapter, KeyAdapter, ComponentAdapter, ContainerAdapter, DragSourceAdapter, DropTargetAdapter, FocusAdapter, WindowAdapter, …
AWT Robot!

• Used to simulate keyboard and mouse programmatically.
• It places events in the native system queues for the platform you are on (not just the java queue).
• Used for recording and replaying activities in regression testing and other uses.
Multithreading and Swing

- Swing components always execute on a single thread within your application. Not the main thread of your application, either.
- Swing components are NOT multithread safe!
- This is done for speed but influences how you must design for them.
- We can ignore this for prototyping UIs but not for design of applications.
Swing Components
Sample dialog with a few controls.

MySampleOfSwingControls1.java
JButton

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- javax.swing.JComponent
        |-- javax.swing.AbstractButton
          |-- javax.swing.JButton
JButton

- Used for a command
- Push and shows a state change visually (pliancy)
- Has a name, label text,
Adding items to the List
Add items to the model for the list
Changing the border of a list box
A Titled Border for a List Box
List Box Selection Modes

- Single
- Multiple_Interval
- Single_Interval
Setting the Button group on a radio button
Setting the Mnemonics
Final Dialog Version

Burger Customization Dialog - Please select the attributes and then press the Order Burger button

- Cheese
- Tomato
- Bacon
- Lettuce

- Rare
- Medium Rare
- Medium
- Well Done

Order Burger
JComboBox

• Two styles in the app – non-editable and editable
• If you use the editable type you should check the input of the user to make sure it is acceptable.
• You can change the style by changing the editable property.
Creating a JavaGUI -> SampleForms -> Application
Menu and Menu item hierarchy that defines the menus for the app
The default menus
Copy and paste a menu into the hierarchy
This shows 2 Edit menus