GUI Programming with Java

Session 7
Frame Based GUI's

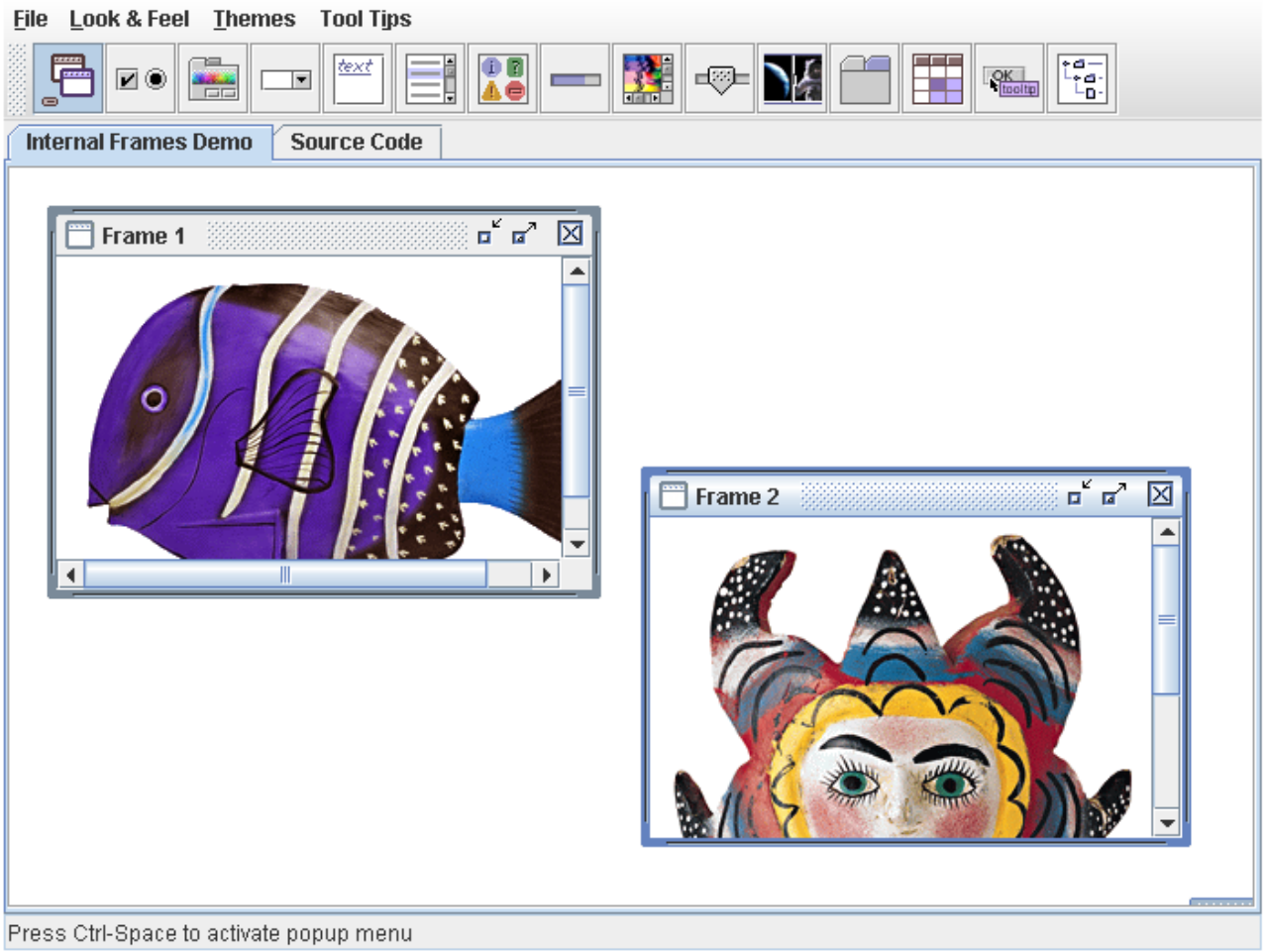
- We will look at…
  - JInternalFrame

- Many GUI's applications simulate desktop environments by allowing multiple frames or windows to be displayed within a single root window.

- These frames look and in many cases behave like those on our normal desktop even though the window manager associated with the desktop is not aware of them (they are internal to the application)

- Typical examples include graphics applications, word processors, office applications etc.

- In this session we will examine the methods for creating internal frame based applications.

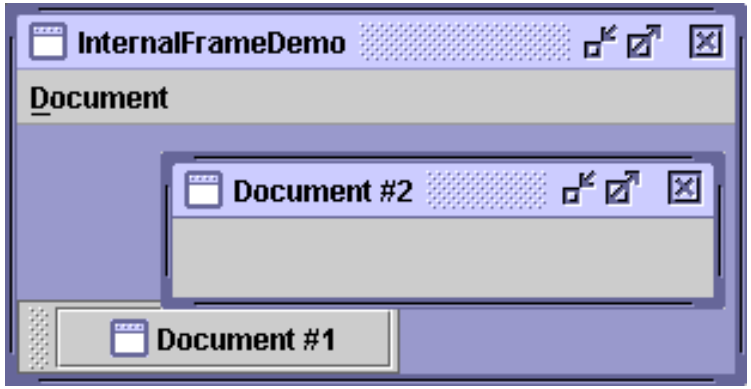


The above screenshot is from the swing demo we looked at in session 1. It makes use of internal frames.

- With the JInternalFrame class you can display a JFrame-like window within another window.

- Usually, you add internal frames to a desktop pane. (using JDesktopPane)

- The desktop pane, in turn, might be used as the content pane of a JFrame.



- Lets now take a look at

  InternalFrameDemo1.java

  which accompanies this lecture.

- We base our class on a JFrame. In other words, our application will consist of a JFrame that will in turn consist of a number of inner frames

- We import some new libraries to allow us to make use of the internal frame and the desktop pane

```
import javax.swing.JInternalFrame;
import javax.swing.JDesktopPane;
import javax.swing.JFrame;
```

- We begin by creating a JDesktopPane object

```
JDesktopPane desktop = new JDesktopPane();
```

- This is simply a pane on which we can add content, only in this case it will be frames.

- We then create an internal frame

```
JInternalFrame innerframe = new JInternalFrame("Internal Window" ,
                                    true, //resizable
                                    true, //closable
                                    true, //maximizable
                                    true);//iconifiable
```

- When creating a frame we specify its title, weather it is resizable, closable, maximizable or iconifiable

- Just like a standard JFrame, we can set the size and visibility and other properties of the JInternalFrame

```
innerframe.setSize(300,300);
innerframe.setVisible(true); //necessary as of 1.3
```

- We then get the desktop to act as the content pane for the JFrame

```
setContentPane(desktop);
```

- Most frame based applications use menu's so lets expand our program and add a menu to it.

- Lets now take a look at

  InternalFrameDemo2.java

  which accompanies this lecture.

- You can see that the menu does not do much.
- However the methods for adding menus are exactly the same as standard frames.
- Remember that our program is based on a JFrame

A more complex example

- Take a look at the program **InternalFrameDemo.java**
- It makes use of another class called **MyInternalFrame.java**
- This example allows us to create a series of new windows using a menu.

Rules of using Internal Frames

- **You must set the size of the internal frame.**
  - If you don't set the size of the internal frame, it will have zero size and thus never be visible. You can set the size using one of the following methods: setSize, pack, or setBounds.

- **As a rule, you should set the location of the internal frame.**
  - If you don't set the location of the internal frame, it will come up at 0,0 (the upper left of its container). You can use the setLocation or setBounds method to specify the upper left point of the internal frame, relative to its container.

Rules of using Internal Frames

- **To add components to an internal frame, you add them to the internal frame's content pane.**
  - This is exactly like the JFrame situation.

- **Dialogs that are internal frames should be implemented using JOptionPane or JInternalFrame, not JDialog.**
  - To create a simple dialog, you can use the JOptionPane showInternal*Xxx*Dialog methods.

Rules of using Internal Frames

- **You must add an internal frame to a container.**
  - If you don't add the internal frame to a container (usually a JDesktopPane), the internal frame won't appear.
- **You need to call show or setVisible on internal frames.**
  - Internal frames are invisible by default. You must invoke setVisible(true) or show() to make them visible.

- **Internal frames fire internal frame events, not window events.**
  - Handling internal frame events is almost identical to handling window events.
  - Visit http://download.oracle.com/javase/tutorial/uiswing/events/internalframelistener.html for more information on internal frame events.

- For more information on internal frames visit

  http://download.oracle.com/javase/tutorial/uiswing/components/internalframe.html#construct

Some useful methods

setJMenuBar
you can add menus to internal frames

setLocation
position the frame

setBounds
Explicitly set the size and location of the internal frame.

setSize
Explicitly set the size of the internal frame.

moveToFront() or moveToBack()
If the internal frame's parent is a layered pane such as a desktop pane, moves the internal frame to the front or back (respectively) of its layer.

Some useful methods

setClosed(boolean)

Set or get whether the internal frame is currently closed. The argument to setClosed must be true.

setFrameIcon(Icon)

Set the icon displayed in the title bar of the internal frame (usually in the top-left corner).

JInternalFrame[] getAllFrames()

Returns all JInternalFrame objects that the desktop contains.

Event Handling for Internal Frames

- An InternalFrameListener is similar to a WindowListener.
- Like the window listener, the internal frame listener listens for events that occur when the "window" has been shown for the first time, disposed of, iconified, deiconified, activated, or deactivated.
- Take a look at InternalFrameEventDemo.java for a sample of how we can handle internal frame events.

Labwork 7

Task 7-1

Write a class that creates a window with an internal frame which has the following properties

- Internal Frame Title: My First Internal Frame
- Size: 640 X 480
- Location: 10 pixels in from the top and left of the parent frame
- Allow Resize: No

Hint: Use Java Documentation

Labwork 7

Task 7-2

Write a class that adds a file menu to the internal frame from Task 7-1.
The menu should have the following items

- New
- Save
- Close

Task 7-3

Write a class that uses a for loop to create five windows on screen.

END OF SESSION