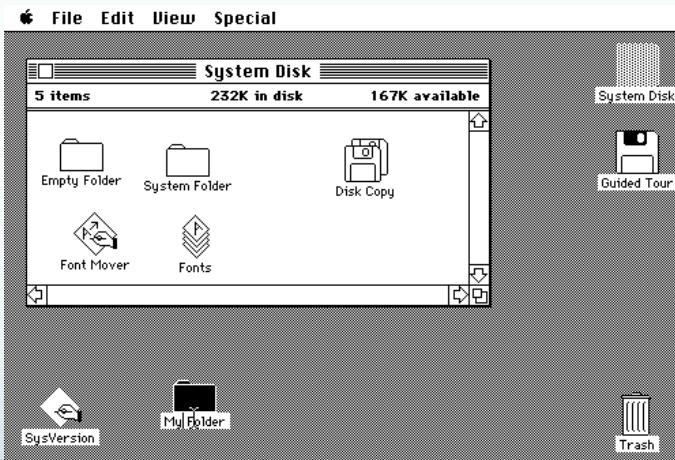


Java GUI Basics

COMP3111/H tutorial

Graphical User Interface (GUI)

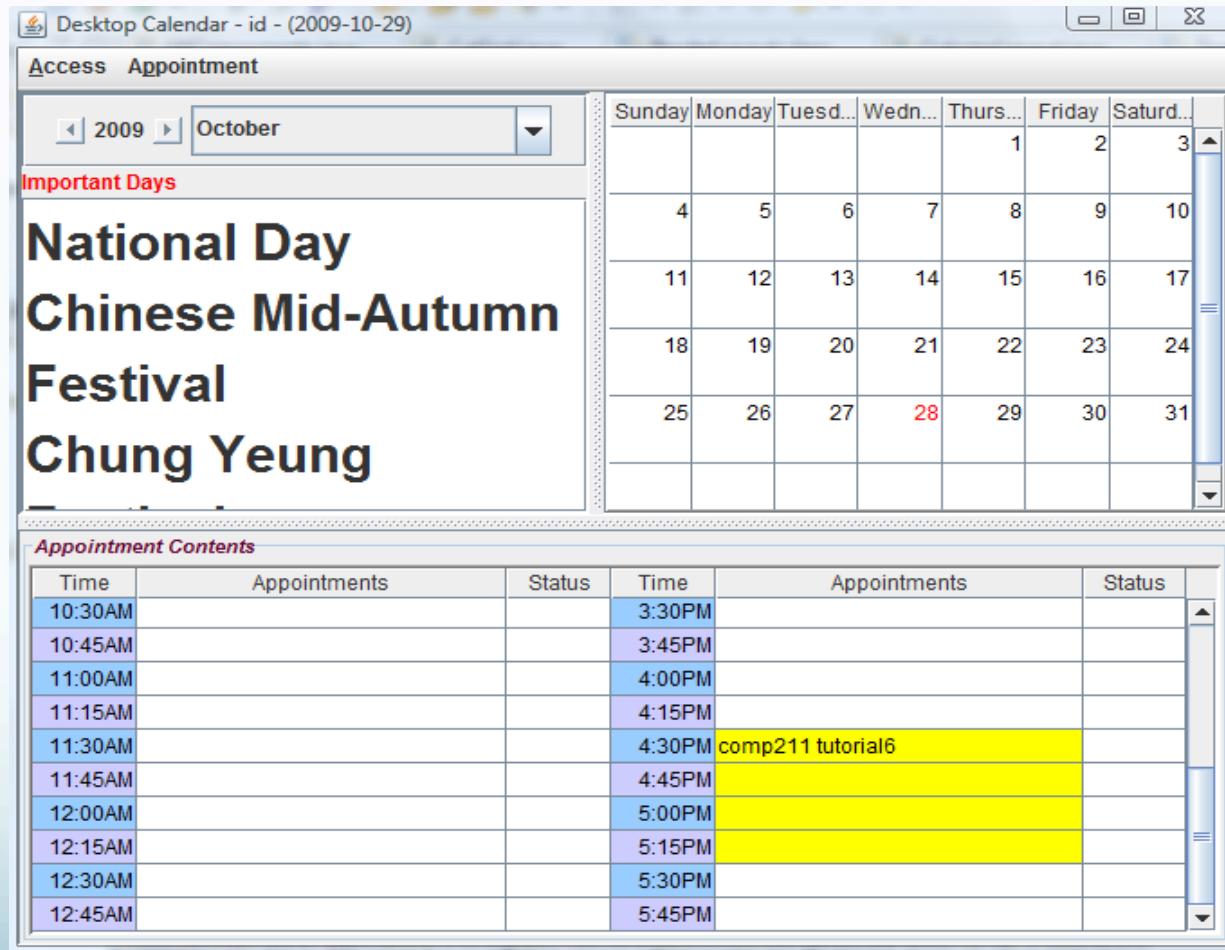
- A **graphical user interface (GUI)** is a type of user interface item that allows people to interact with programs in more ways than typing
 - Compare: Command-line interface



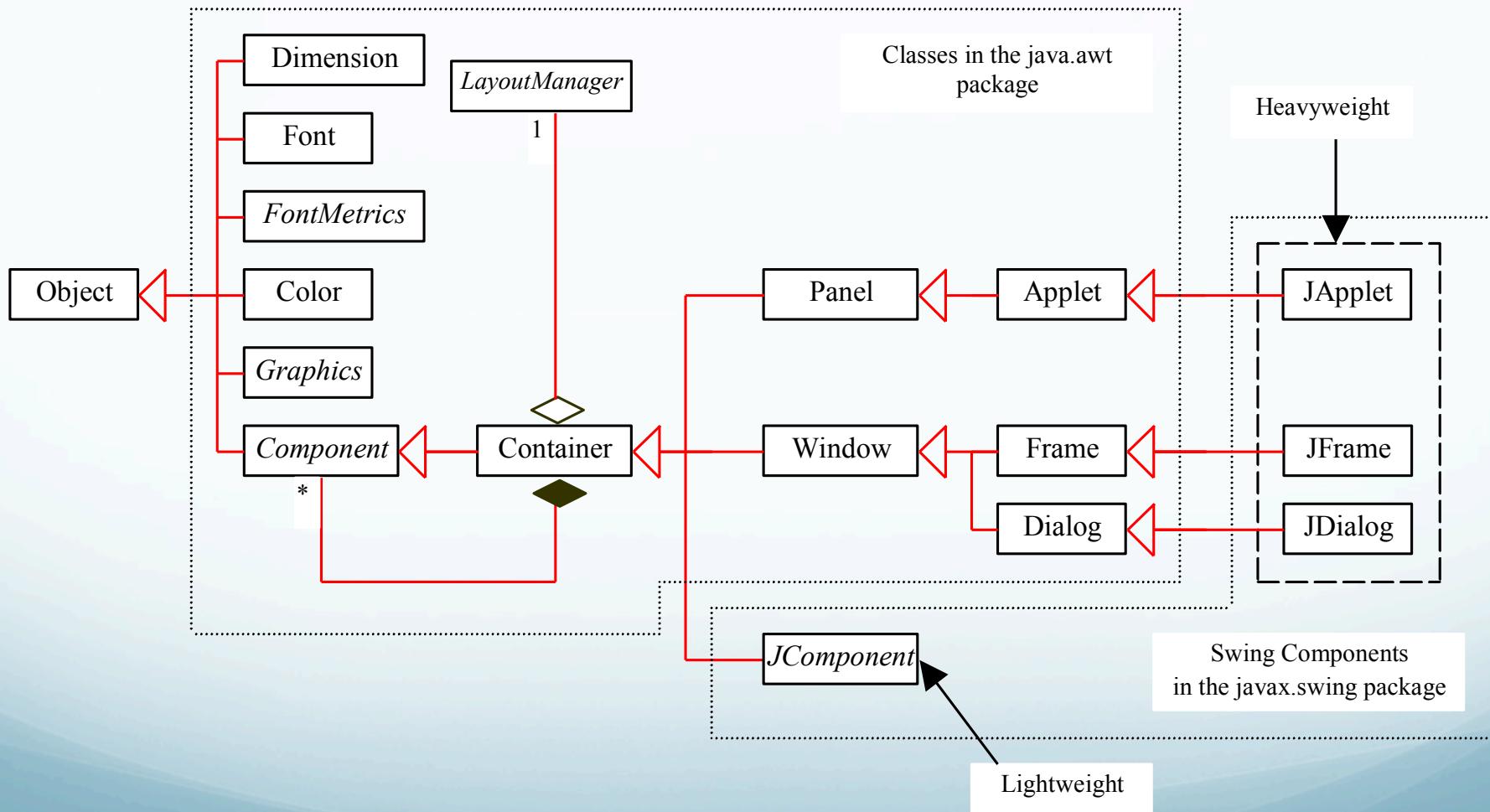
Java GUI Frameworks

- AWT (Abstract Window Toolkit)
- Swing
 - distributed with the Java release (i.e. no external library is necessary)
- SWT (Standard Widget Toolkit)
 - a platform-specific SWT library (JAR file) must be distributed with each application
 - SWT widgets have the same "look and feel" as native widgets
 - SWT is used to develop Eclipse IDE

Calendar GUI by Swing

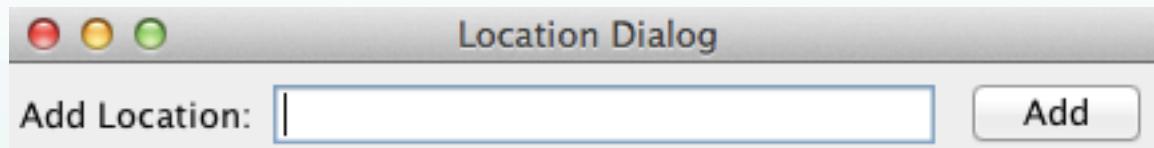


Swing Hierarchy



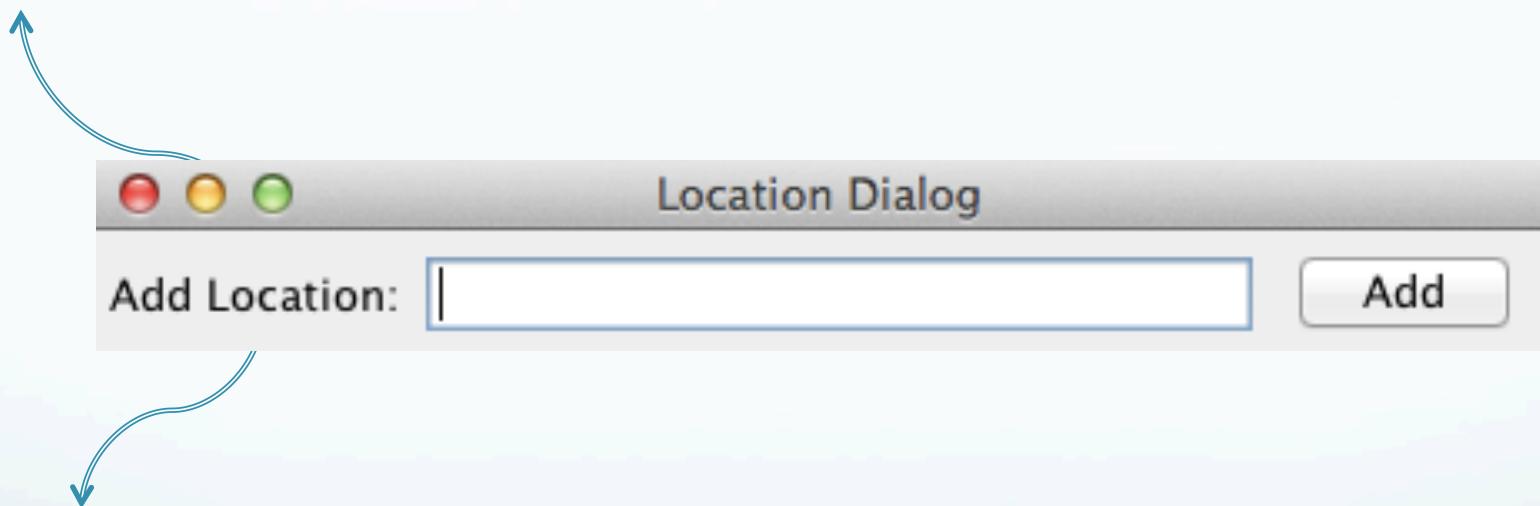
Learn Swing by Example

- How can we build the following user interface using Swing?
- What Swing classes we should use to build up this user interface?
 - Hint: Swing classes usually start with letter 'J'



Learn Swing by Example

JFrame: a top-level window with a title and a border



JPanel: serves as a container of 3 objects: JLabel, JTextField, and JButton

Step 1: LocationDialog Declaration

```
1①import java.awt.*;           → For convenience,  
2import javax.swing.*;  
3  
4public class LocationDialog extends JFrame {  
5  
6    private JLabel lblAddLocation;  
7    private JTextField txtAddLocation;  
8    private JButton btnAddLocation;  
9}
```

To store
3 Swing objects

Step 2: Construct the GUI inside the constructor

```
public LocationDialog() {  
    setTitle("Location Dialog");  
    JPanel contentPane = new JPanel();  
    lblAddLocation = new JLabel("Add Location:");  
    txtAddLocation = new JTextField(20); // 20 characters  
    btnAddLocation = new JButton("Add");  
    contentPane.add(lblAddLocation);  
    contentPane.add(txtAddLocation);  
    contentPane.add(btnAddLocation);  
    setContentPane(contentPane); // set the contentPane  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // close on exit  
    pack(); // resize to fit all GUI components  
    setVisible(true); // make the dialog visible  
}
```

Step 3: Create the top-level container object

- It is a recommended practice in use the “invokeLater” utility method to create an instance of top-level Swing component

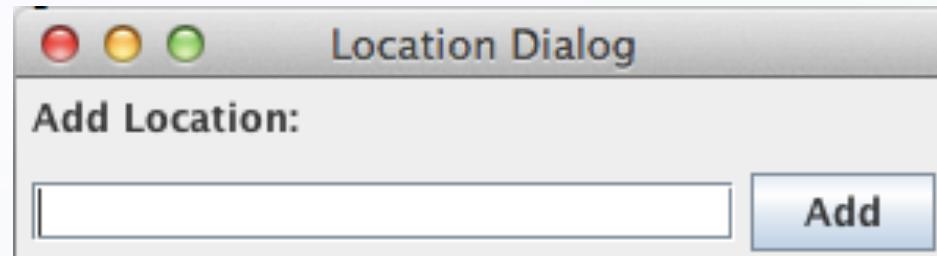
```
public class DemoSwing {  
    public static void main(String[] args) {  
        javax.swing.SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                // Create a new Location dialog  
                LocationDialog dlg = new LocationDialog();  
            }  
        });  
    }  
}
```

Pluggable Look-and-feel

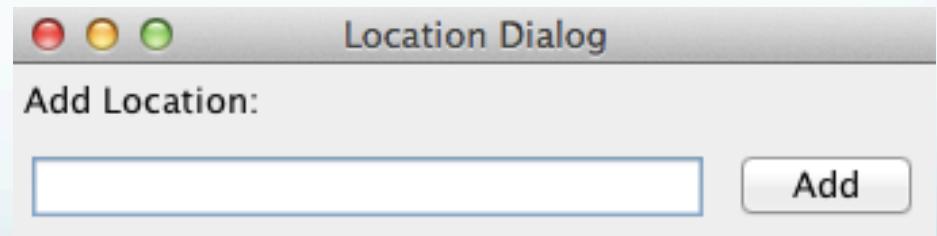
- Change the look-and-feel of Swing by a Java programming statement
- Possible values:
 - com.sun.java.swing.plaf.motif.MotifLookAndFeel
 - com.sun.java.swing.plaf.windows.WindowsLookAndFeel
 - javax.swing.plaf.metal.MetalLookAndFeel

Pluggable Look-and-feel

- Metal (Default)

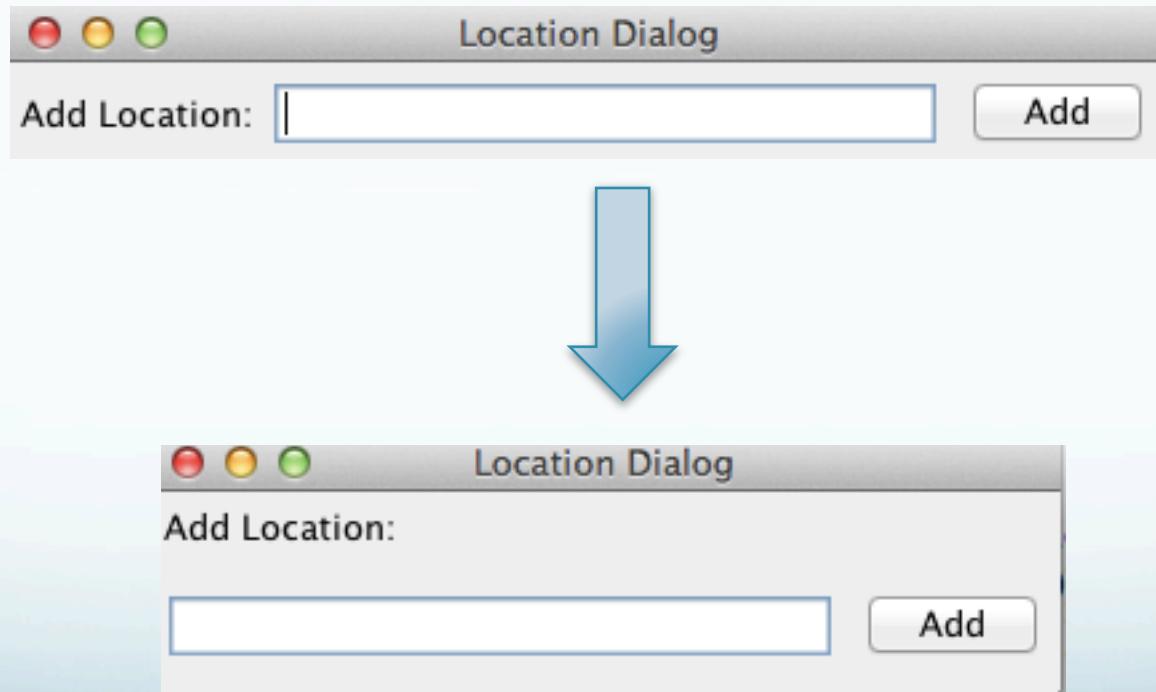


- Windows look-and-feel



Refine the GUI design

- How can we refine the user interface?



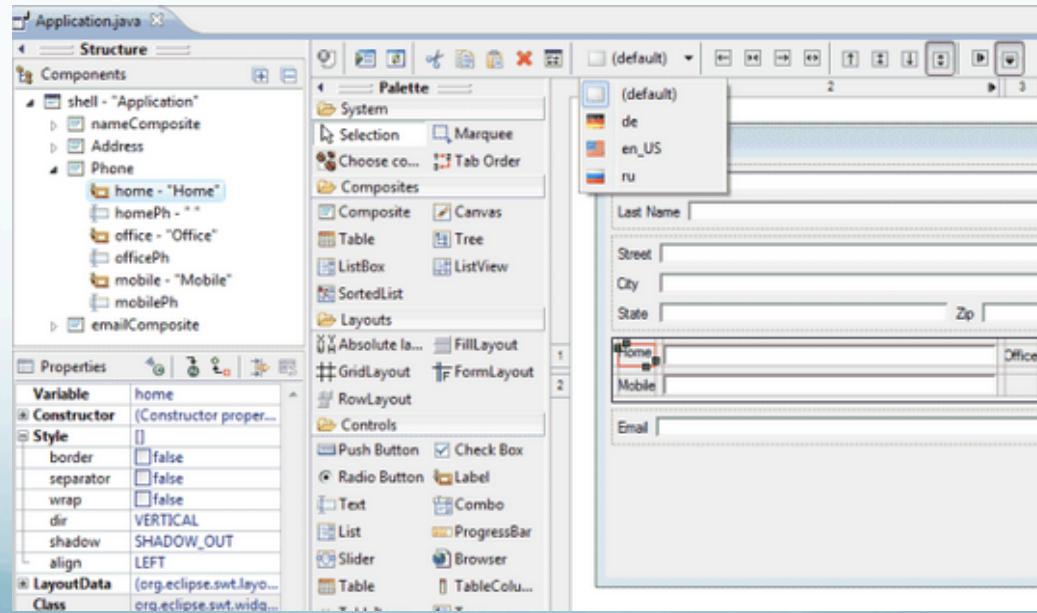
Using Layout manager

- Using Layout manager such as FlowLayout and BoxLayout

```
JPanel top, bottom ; // introduce 2 intermediate panels  
top = new JPanel();  
top.setLayout(new FlowLayout(FlowLayout.LEFT));  
top.add(lblAddLocation);  
contentPane.add(top);  
bottom = new JPanel();  
bottom.add(txtAddLocation);  
bottom.add(btnAddLocation);  
contentPane.add(bottom);  
contentPane.setLayout(new BoxLayout(contentPane, BoxLayout.Y_AXIS));
```

WYSIWYG editor

- Coding a complex Swing GUI by hand-written Java programming statements might be tedious
- You may consider to use a WYSIWYG editor
 - Example: <http://www.eclipse.org/windowbuilder/>



Event handling

- Nothing happens when I click the “Add” button
- Can we do something interesting (e.g. show a message box if the “Add” button is clicked)?



ActionListener

- JButton has a method called addActionListener
- By using the syntax of Anonymous Class, an appropriate action (e.g. show a message dialog) can be triggered

```
btnAddLocation.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "The Add button is clicked!");
    }
});
```

Self-learning by reading online tutorials

- Java GUI programming can't be taught in a single lesson. Self-learning is necessary
- Online tutorials / references:
 - <http://docs.oracle.com/javase/tutorial/uiswing/>
 - <http://www.tutorialspoint.com/swing/>
- Tutorial on WindowBuilder (WYSIWYG editor)
 - <http://users.ece.utexas.edu/~miryung/teaching/EE379K-Spring2014/SwingTutorial.html>

Self-learning by reading Java Documentation

- Example: Search “JTable java” in Google



A screenshot of a Google search results page. The search query "JTable java" is entered in the search bar. The "Web" tab is selected, showing approximately 490,000 results found in 0.20 seconds. The top result is a link to the Oracle Java Platform SE 7 documentation for JTable, located at docs.oracle.com/javase/7/docs/api/javax/swing/JTable.html. The snippet describes JTable as a tool for displaying and editing two-dimensional tables of cells. Below the snippet are links to related classes: TableModel, TableColumnModel, TableColumn, and JTableHeader. At the bottom of the snippet, it indicates the user has visited the page 2 times and their last visit was on 9/18/14.

About 490,000 results (0.20 seconds)

JTable (Java Platform SE 7) - Oracle Documentation
docs.oracle.com/javase/7/docs/api/javax/swing/JTable.html ▾

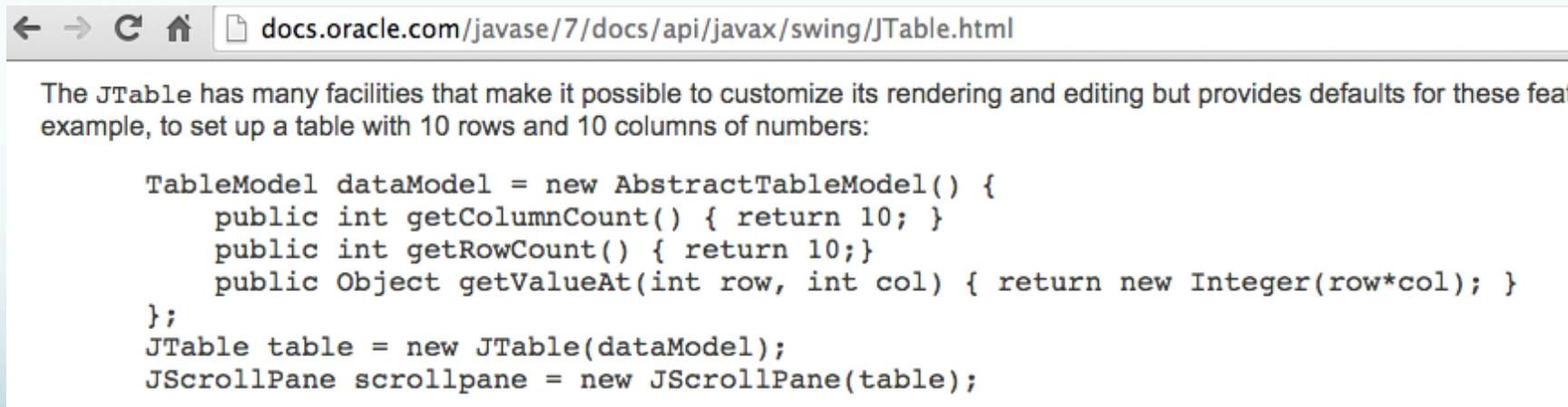
The **JTable** is used to display and edit regular two-dimensional tables of cells. See How to Use Tables in The Java Tutorial for task-oriented documentation and ...

[TableModel](#) - [TableColumnModel](#) - [TableColumn](#) - [JTableHeader](#)

You've visited this page 2 times. Last visit: 9/18/14

Self-learning by reading Java Documentation

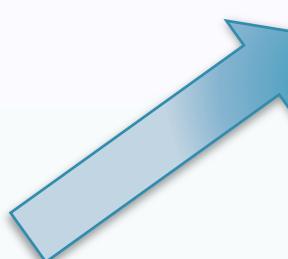
- Usually, code snippet will be given in the online documentation
- You may first copy and paste to a Java project, and change the values and/or method calls



The screenshot shows a web browser window displaying the Java API documentation for `JTable`. The URL in the address bar is `docs.oracle.com/javase/7/docs/api/javax/swing/JTable.html`. The page content describes the `JTable` class and provides a code snippet for creating a table with 10 rows and 10 columns of numbers.

```
TableModel dataModel = new AbstractTableModel() {
    public int getColumnCount() { return 10; }
    public int getRowCount() { return 10; }
    public Object getValueAt(int row, int col) { return new Integer(row*col); }
};
JTable table = new JTable(dataModel);
 JScrollPane scrollpane = new JScrollPane(table);
```

Result



D	0	0	0	0	0	0	0	0	0	0
D	1	2	3	4	5	6	7	8	9	
D	2	4	6	8	10	12	14	16	18	
D	3	6	9	12	15	18	21	24	27	
D	4	8	12	16	20	24	28	32	36	
D	5	10	15	20	25	30	35	40	45	
D	6	12	18	24	30	36	42	48	54	
D	7	14	21	28	35	42	49	56	63	
D	8	16	24	32	40	48	56	64	72	
D	9	18	27	36	45	54	63	72	81	

```
TableModel dataModel = new AbstractTableModel() {
    public int getColumnCount() { return 10; }
    public int getRowCount() { return 10; }
    public Object getValueAt(int row, int col) { return new Integer(row*col); }
};
JTable table = new JTable(dataModel);
 JScrollPane scrollpane = new JScrollPane(table);

contentPane.add(table); // add the table to the content pane
```