

Swing

Swing \subset JFC

- Java Foundation Classes
 - Swing Components
 - Java2D API
 - Drag & drop
- Swing can be added to AWT in JDK 1.1

2

Swing vs. AWT

java.awt.Container
java.awt.Frame

- javax.swing.*
- Components renamed: JComponent, JButton, JPanel, JFrame...
- getContentPane () in JFrame, JDialog, JApplet
- paintComponent instead of paint
- New components: JTable, JTree, MDI windows
- Native code moved to Java code (lightweight) \Rightarrow more functionality
 - Images on JLabels and JButtons
 - Borders
 - Paint on predefined components
 - Components need not be rectangular (can be e.g. circular)
 - Set look & feel
- MVC for components with state: JList, JTable, JTree, JEditorPane
- Do not mix AWT and Swing components in the same UI

3

Swing Components

4

Borders

5

HTML in Component Labels

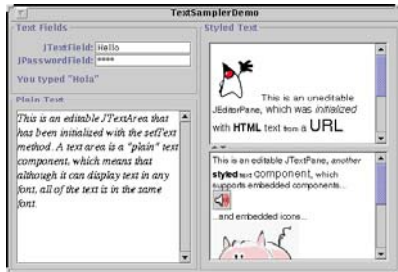
- Buttons, labels, menu items, tool tips, tabbed panes, trees, tables

```
new JButton ("
    <html>
        <b><u>T</u>wo</b><br>
        lines
    </html>
")
```

- Cannot be used in text fields or text areas
- For editable HTML code, use editor pane

6

Text, HTML, and RTF Editors javax.swing.JEditorPane



7

JTable

First Name	Last Name	Sport	# of Years	Vegetarian
Mark	Andrews	Speed reading	20	<input type="checkbox"/>
Way	Campione	Snowboarding	5	<input type="checkbox"/>
Alison	Huml	Rowing	3	<input checked="" type="checkbox"/>
Angela	Lih	Teaching high...	4	<input type="checkbox"/>

- Control over file and column size
- Selectable cells, rows, columns
- Editable cells, with validation
- Table sorting
- Custom cell rendering
- Custom cell editors

8

Model View+Controller: Example

```
package javax.swing.table.*;

public interface TableModel {
    int getColumnCount ()
    int getRowCount ()
    String getColumnName (int col)
    Object getValueAt (int row, int col)
    void setValueAt (Object value, int row, int col)
    boolean isCellEditable (int rowIndex, int columnIndex)
    void addTableModelListener (TableModelListener l)
    void removeTableModelListener (TableModelListener l)
}
```

9

Model View+Controller: Example

```
import javax.swing.table.*;

class MyModel extends AbstractTableModel {
    private String[] colNames = { "Name", "Birth" };
    private Object[][] data = { {"John", "1/2/1980"},
                                {"Patricia", "3/4/1979"} };
    public int getColumnCount() { return colNames.length; }
    public int getRowCount() { return data.length; }
    public String getColumnName (int col) { return colNames[col]; }
    public Object getValueAt (int row, int col) {
        return data[row][col];
    }
    public Class getColumnClass (int c) {
        return getValueAt (0, c).getClass();
    }
}

// Somewhere else, building a UI...
panel.add (new JTable(new MyModel ());)
```

10

New Events

- DocumentEvent
- TableModelEvent
- TreeModelEvent
- TreeExpansionEvent
- TreeSelectionEvent
- ListDataEvent
- ListSelectionEvent

11

Predefined Dialog Boxes javax.swing.JOptionPane

```
JOptionPane.showMessageDialog (null, "Aviso", "Alert",
                                JOptionPane.ERROR_MESSAGE);
```



```
JOptionPane.showConfirmDialog (null, "Elige uno", "Choose one",
                                JOptionPane.YES_NO_OPTION);
```




```
String inputValue = JOptionPane.showInputDialog ("Please input a value");
```




12

```
Object[] options = { "OK", "CANCEL" };
JOptionPane.showOptionDialog (null, "Click OK to continue", "Warning",
    JOptionPane.DEFAULT_OPTION,
    JOptionPane.WARNING_MESSAGE,
    null, options, options[0]);
```



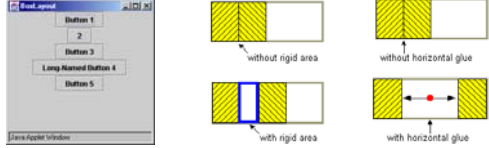
```
Object[] possibleValues = { "First", "Second", "Third" };
Object selectedValue =
    JOptionPane.showInputDialog (null, "Choose one", "Input",
        JOptionPane.INFORMATION_MESSAGE,
        null, possibleValues, possibleValues[0]);
```



13

javax.swing.BoxLayout

- Row or column layout
- Preserves preferred dimensions (maximum/minimum)
- Lets each component decide its alignment using `setAlignmentX`, `setAlignmentY`
- Admits invisible components, rigid or flexible (glue)



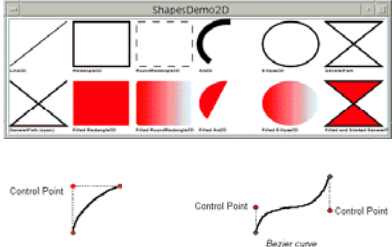
14

Java2D API The java.awt.geom package

- Geometric forms with self-drawing capability
 - 2D shapes: `Line2D`, `Rectangle2D`, `RoundRectangle2D`, `Arc2D`, `Ellipse2D`, `QuadCurve2D`, `CubicCurve2D`, `GeneralPath`
 - `java.awt.Graphics2D` and `draw(Shape)`
- Line thickness
- Color effects: gradient, texture
- Move, rotate, scale, shear text and graphics
- Intersection of geometric forms
- Image filters: blur, sharpen
- Contains method for interaction
- Printing

15

2D Shapes




See demo at <http://java.sun.com/docs/books/tutorial/2d/display/strokeandfill.html>

16

Example: GeneralPath

```
public void paintComponent (Graphics g) {
    GeneralPath oddShape = new GeneralPath();
    int x0 = 150, y0 = 25, x, y;
    x = x0; y = y0; oddShape.moveTo (x, y);
    x -= 100;    oddShape.lineTo (x, y);
    y += 50;    oddShape.lineTo (x, y);
    x += 100;   oddShape.lineTo (x, y);
    oddShape.curveTo (x + 10, y - 10, x - 10, y - 30, x0, y0);
    Graphics2D g2 = (Graphics2D)g;
    g2.draw (oddShape);
}
```




17

Java2D API: More Features

Clip **The Starry Night**

Shear **Text**

Composition



18