



程式語言 (Programming Language)

Lec. 05 圖形使用者介面 (GUI) I



楊 吳 泉

義守大學資訊工程學系

web: <http://elearning.isu.edu.tw>

<http://audtm.net/~wcyang>

mail: wcyang@isu.edu.tw

綱要 (Outline)

➤ GUI觀念

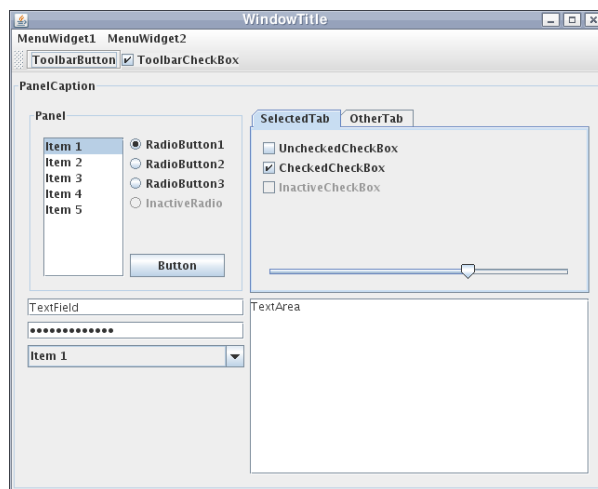
✓ GUI: 圖形使用者介面(Graphic User Interface, GUI)

➤ 常見元件(Component)

➤ 版面配置(Layout)

➤ 事件處理(Event Handling)

➤ 程式範例與練習





GUI觀念

➤ 使用者介面(User Interface, UI)

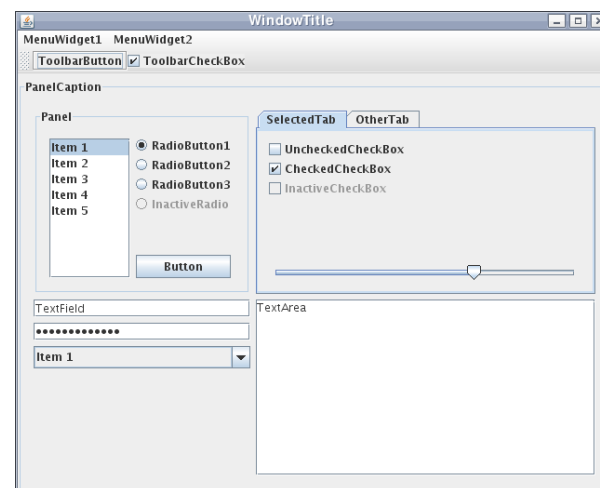
✓ 文字介面

- 命令列模式 (Command mode)
- 節省系統資源

✓ 圖形介面

- 友善容易操作

```
D:\eclipse_java\workspace\HexInteger\bin>java HexPrint2
Input an Integer: 345678
a(DEC) = 345678
a(hex) = 5464e
a(HEX) = 5464E
```



➤ 圖形使用者介面(Graphic User Interface, GUI)

✓ 由使用者依據所見選擇所需要操作

- 所見: 圖形元件, 如選單(Menu)、按鍵(Button)
- 操作: 事件處理, 需撰寫對應操作之程式



Java GUI

➤ 組成

✓ Component

- 畫面的主要組成部分
- Java本身並沒有可見的實體, 其子類別才有顯現的效果
- 可以放入其他的Component者, 稱為Container

✓ LayoutManager

- Container內Component的排放位置, 可以用絕對座標, 也可以透過LayoutManager加以調整。

✓ Listener

- 以Mouse, Keyboard操作GUI時, 會產生相對應的Event, 並由視窗系統將該Event傳遞給程式設計者所指定的Listener
- 例如: 使用者以Mouse的左鍵按下Button時, 視窗系統會將傳給對等Listener(Button為ActionListener), 由其決定採取的步驟。
- 這種由使用者觸發的執行動作, 和程序式語言由主程式開始呼叫其他程序的模式不同, 我們稱之為事件驅動(Event Driven)模式。

常見元件(Component)

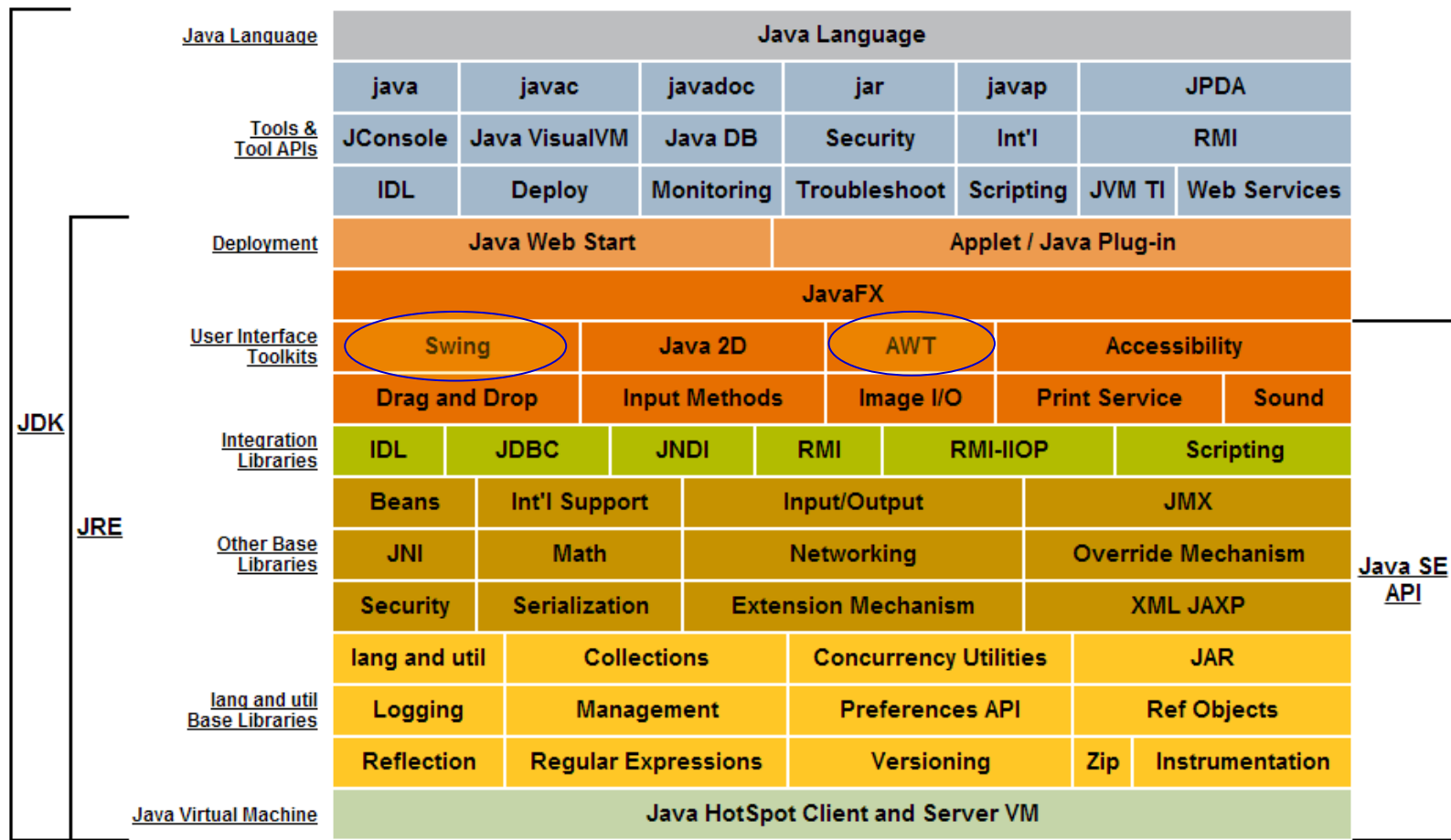
➤ 主要程式套件(Main package)

- ✓ Java API 中有兩個圖形介面套件，分別是 AWT 與 Swing
- ✓ AWT (Abstract Window Toolkit)
 - Java 1.1 版之前官方GUI程式庫 (library) 套件
 - AWT 採用作業系統 (operating system) 提供的視窗元件，所以所設計的視窗會跟作業系統原本的視窗風格一致
- ✓ Swing
 - Java 1.2 版以後又採納 Swing 當第二個官方 GUI 的程式庫
 - Swing 利用 Java 原生程式碼重新繪出視窗元件，因此 Swing 設計的程式在各平台間並無外觀上的差異



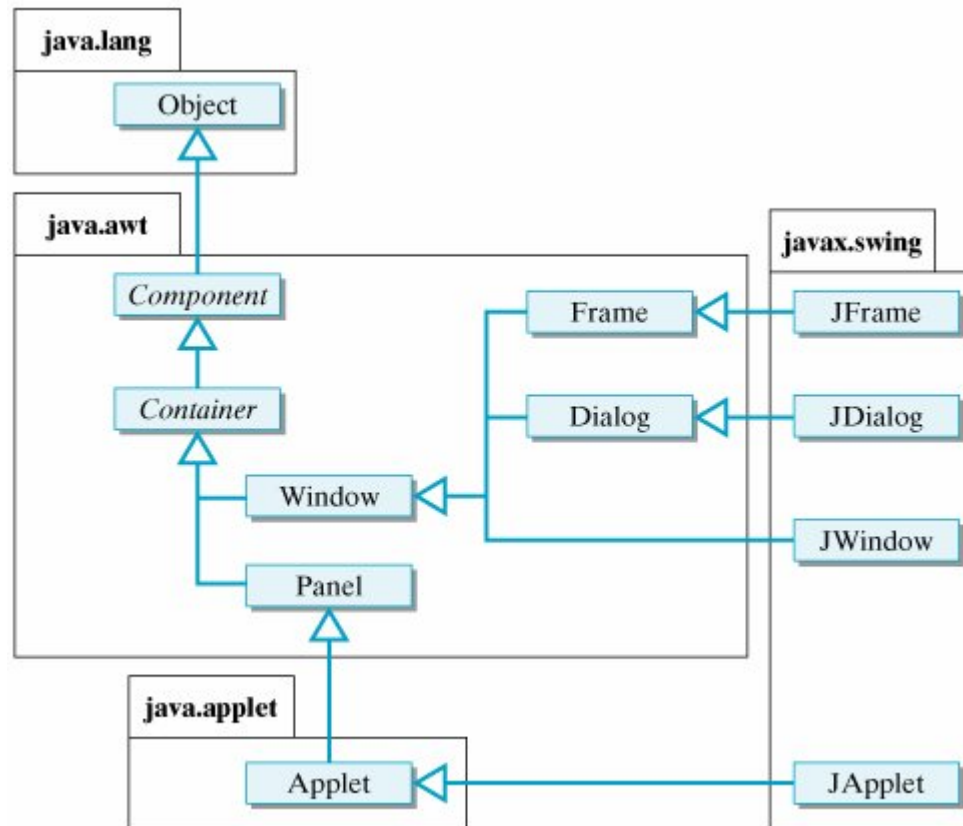


Java GUI in JDK





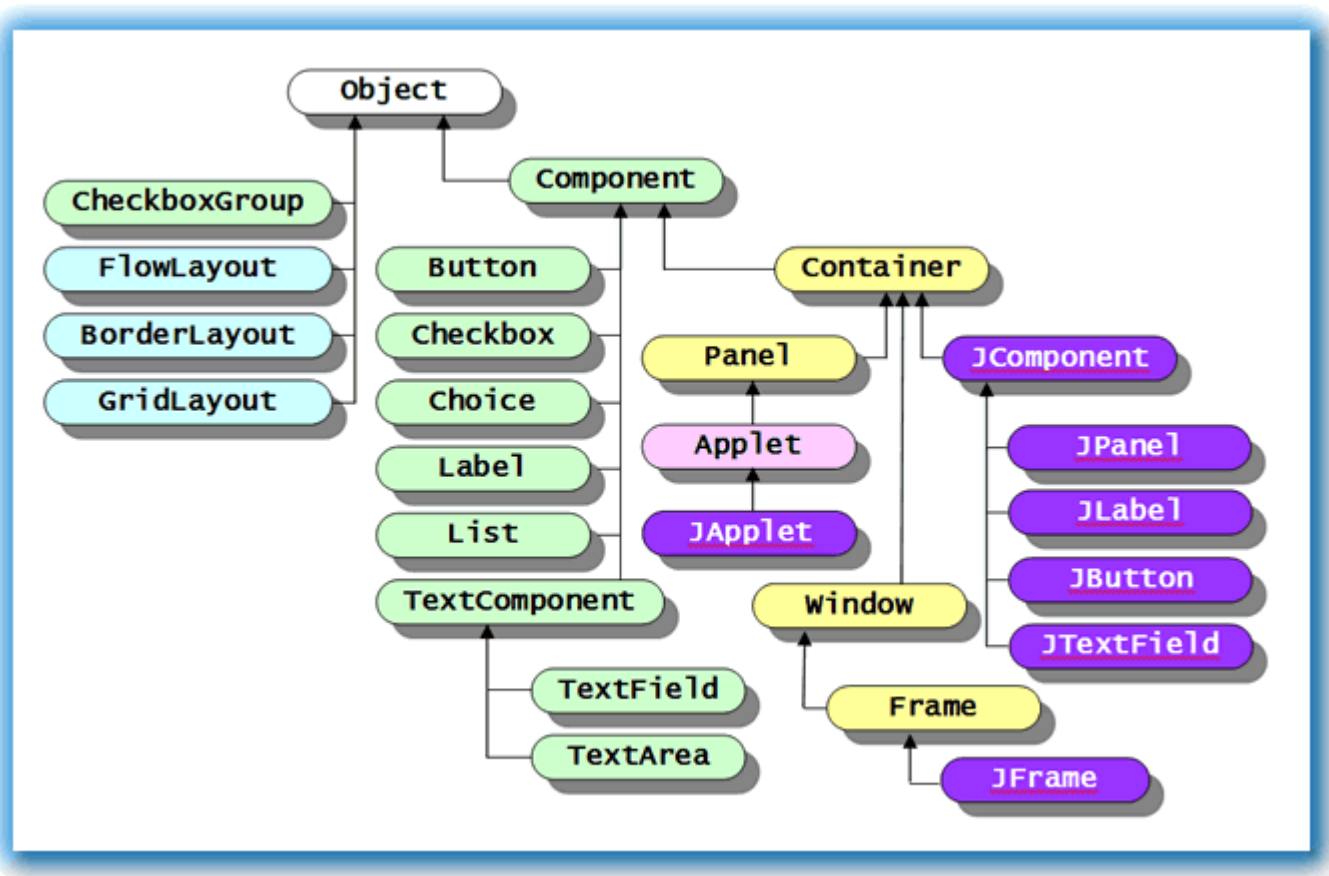
AWT、Swing & Applet



http://www.jroller.com/DhilshukReddy/entry/java_swing_components_hierarchy



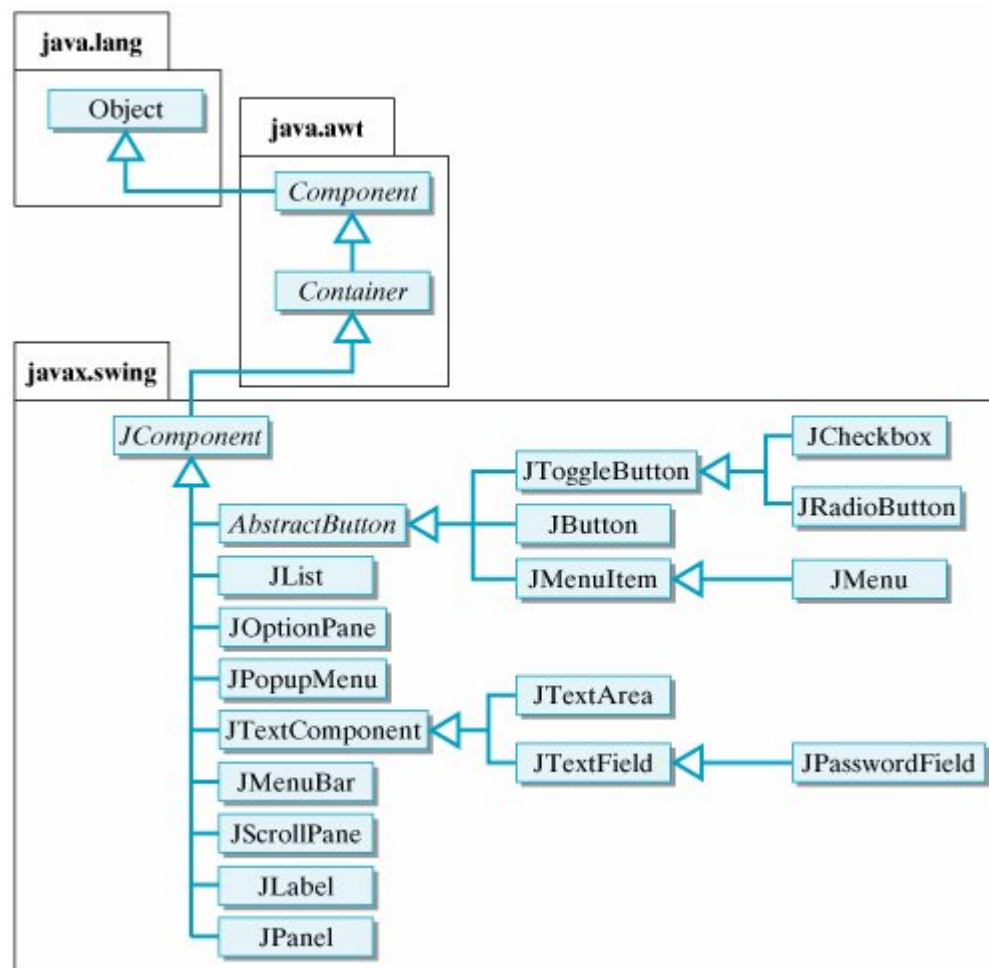
主要AWT元件



<http://faculty.orangecoastcollege.edu/sgilbert/book/08-6-MeetTheWidgets/index.html>



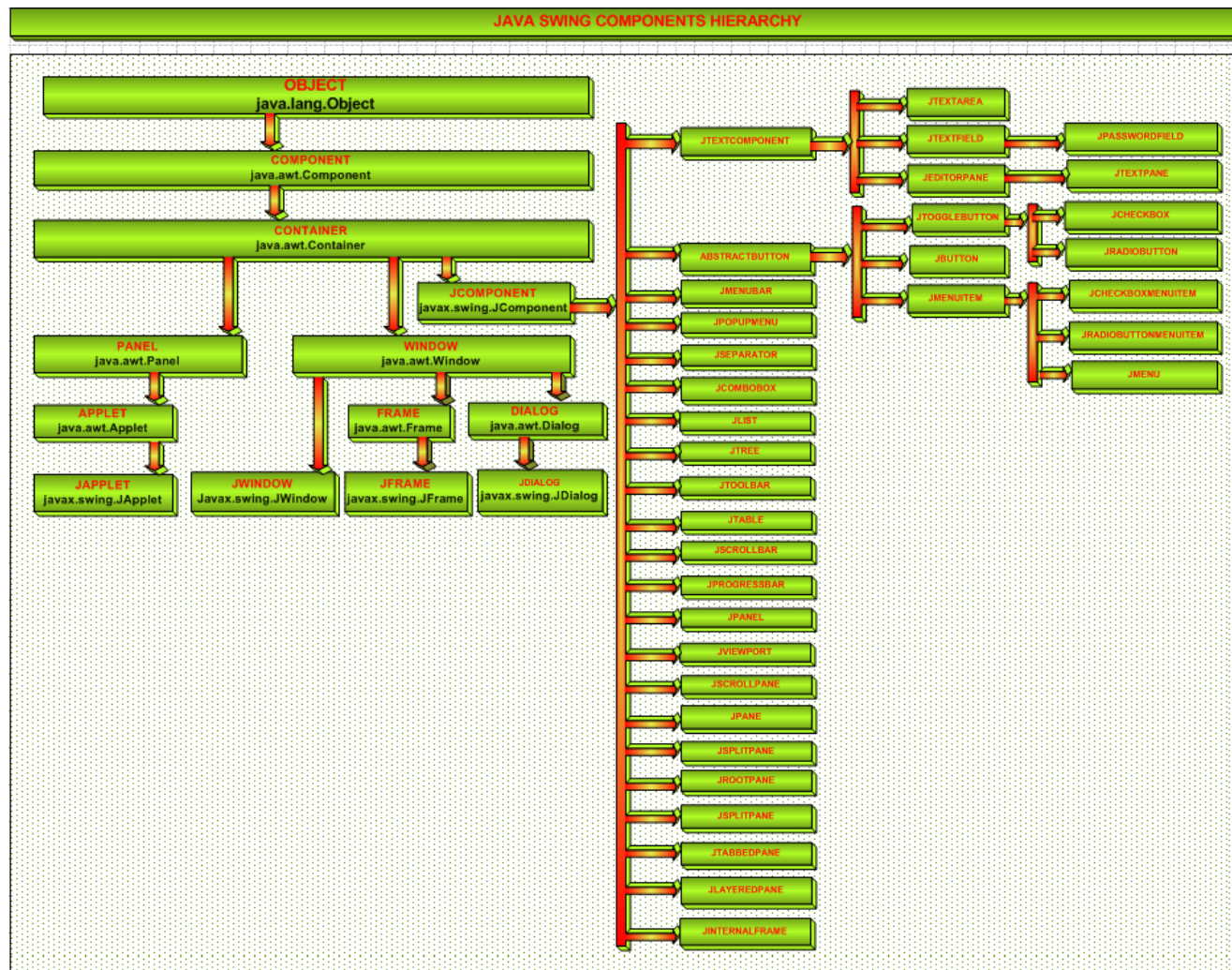
常見Swing 元件



http://www.jroller.com/DhilshukReddy/entry/java_swing_components_hierarchy



Swing 元件架構

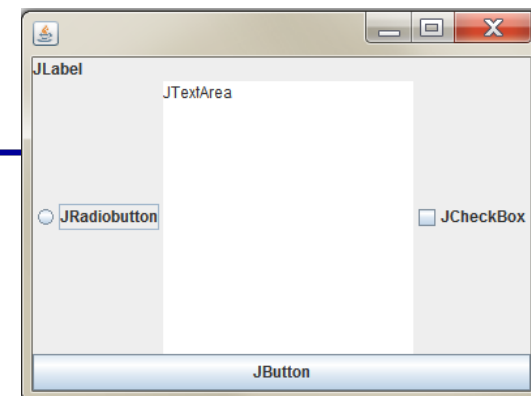


http://www.jroller.com/DhilshukReddy/entry/java_swing_components_hierarchy



程式範例

```
1 import java.awt.*;
2 import javax.swing.*;
3
4 public class Demo01 {
5     public static void main(String[] args) {
6         JFrame demo = new JFrame();
7         demo.setSize(400, 300);
8         demo.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9
10        JCheckBox checkbox = new JCheckBox("JCheckBox");
11        JRadioButton radiobutton = new JRadioButton("JRadiobutton");
12        JButton button = new JButton("JButton");
13        JLabel label = new JLabel("JLabel");
14        JTextArea textarea = new JTextArea("JTextArea");
15
16        demo.getContentPane().add(BorderLayout.EAST, checkbox);
17        demo.getContentPane().add(BorderLayout.WEST, radiobutton);
18        demo.getContentPane().add(BorderLayout.SOUTH, button);
19        demo.getContentPane().add(BorderLayout.NORTH, label);
20        demo.getContentPane().add(BorderLayout.CENTER, textarea);
21
22        demo.setVisible(true);
23    }
24 }
```





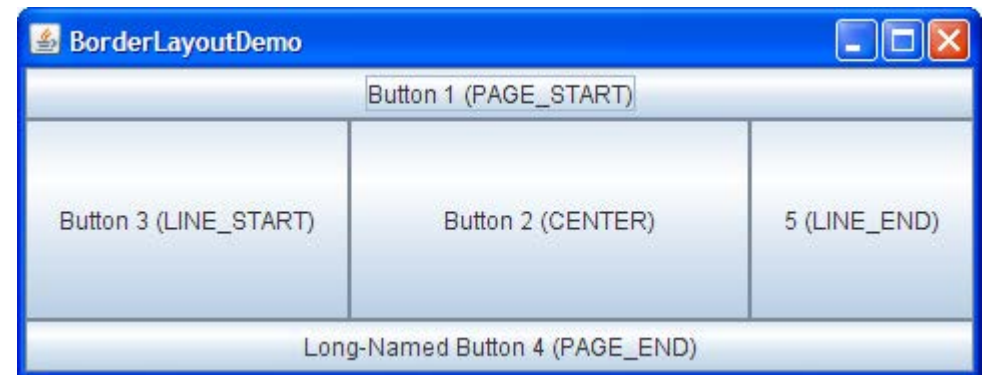
版面配置(Layout)

➤ Layout

- ✓ 元件呈現方式
- ✓ AWT及Swing提供之Layout方式
(<http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>)
 - BorderLayout
 - BoxLayout
 - CardLayout
 - FlowLayout
 - GridBagLayout
 - GridLayout
 - GroupLayout
 - SpringLayout

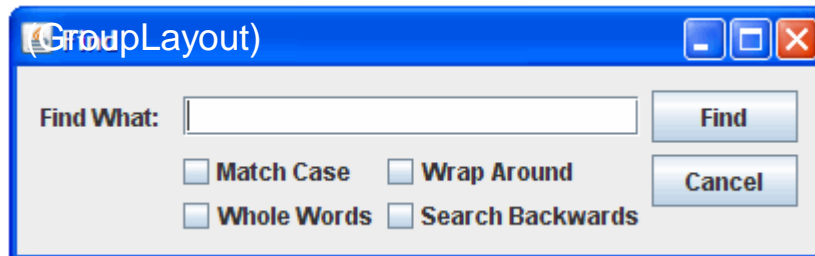
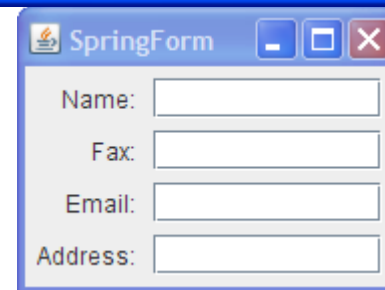
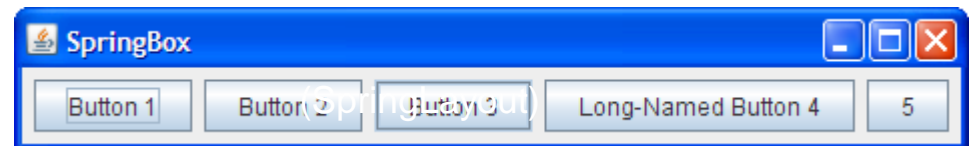
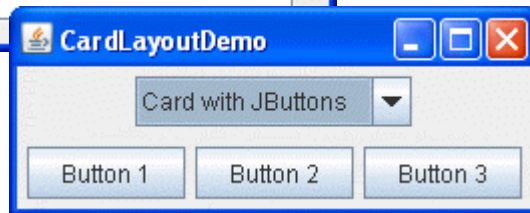
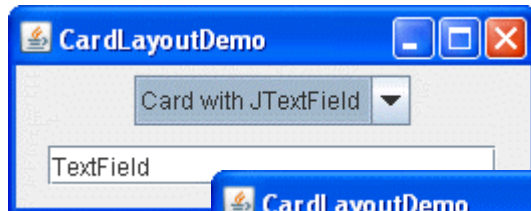
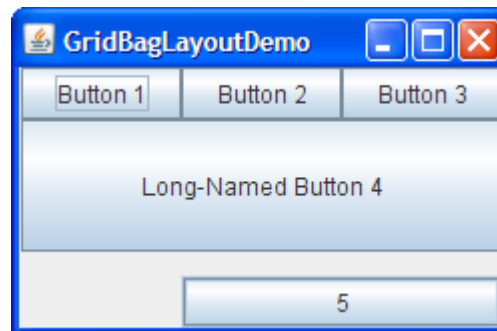
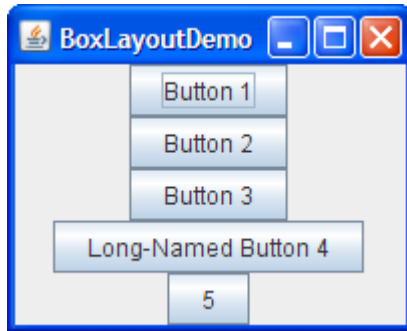


FlowLayout, GridLayout & BorderLayout





Other Layouts

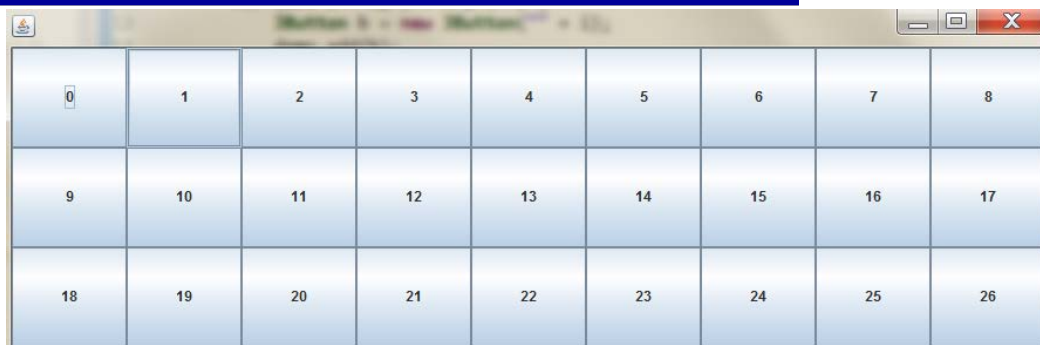




程式範例與練習

➤ 按鍵顯示

✓ 顯示如右之3*9按鍵



```
1 import java.awt.*;
2 import javax.swing.*;
3
4 public class Demo02 {
5     public static void main(String[] args) {
6         JFrame demo = new JFrame();
7         demo.setSize(400, 300);
8         demo.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9         demo.getContentPane().setLayout(new FlowLayout());
10
11         for (int i = 0; i < 27; i++) {
12             JButton b = new JButton("" + i);
13             demo.add(b);
14         }
15
16         demo.setVisible(true);
17     }
18 }
19
20
```

