

Swing (一)

人工智慧與無線感應設備開發專班

湜憶電腦知訊顧問股份有限公司

馬傳義

前言

- GUI（Graphics User Interface，圖形使用者介面使用者界面）是目前程式設計的主流。
- 用Java來設計圖形介面的程式，有AWT與Swing。
 - ◆ AWT（Abstract Window Toolkit，抽象視窗工具）為Java 1.1版之前提供的套件。
 - AWT採用作業系統（Operating System）提供的視窗元件，所以利用 AWT 所設計的視窗圖形程式會跟作業系統原本的視窗風格一致。

前言

- ◆ Swing為Java 1.2版以後提供的套件。
 - Swing是在AWT基礎上構建的。
 - 是利用Java原生程式碼重新繪出視窗元件，因此利用Swing設計的程式在各平台間並無外觀上的差異。
 - Swing套件的特點歸納如下：
 - ◆ 執行階段可更換外觀，或是重新實作元件外觀。
 - ◆ 可利用滑鼠作拖放動作。
 - ◆ 具有提示文字的功能。
 - ◆ 元件較為容易延伸，並創造出自訂的元件。
 - ◆ 支援特定的除錯功能，並提供元件慢動作執行等。

Swing

- Swing是JFC（Java Foundation Classes，Java基礎類別）的一部份，為Java設計GUI（Graphical User Interface，圖形使用者界面）的重要套件。
- swing提供建構視窗程式的各種元件，像是「Label（標籤）」、「Text Field（文字欄位）」、「Button（按鈕）」、「Check Box（核取方塊）」、「Radio Button（選擇按鈕）」、「Combo Box（下拉式清單）」、...等，且保證其跨平台的特性。
- 由於Swing的功能強大，套件內之元件繁多，在此，只針對Swing最新而且最常用的元件來作介紹。

JFrame

- 要製作一個Swing視窗應用程式界面，首先要運用JFrame（視窗）類別來建立一個視窗框架（容器）物件。
- JFrame類別被定義在javax.swing套件之中，所以在程式一開始，必需運用import關鍵字來匯入javax.swing套件及其中所有的類別：

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

- 再透過new指令建立視窗物件。

```
JFrame 物件名稱 = new JFrame ( );
```

JFrame

- 若要自定視窗類別則要用 extends 關鍵字來繼承 JFrame 類別。

```
public class 類別名稱 extends JFrame  
{  
    程式敘述  
}
```

JFrame

◆ JFrame類別的建構子：

- JFrame()
 - ◆ 用來建立一個沒有標題列文字的新視窗物件（容器）。
- JFrame(String title)
 - ◆ 用來建立一個有標題列文字的新視窗物件（容器）。
 - ◆ 引數title為視窗標題列的文字。
 - ◆ 若要用super(String title);敘述來設定視窗標題文字時，要將此敘述放在建構子敘述區塊的第一行。

◆ JFrame類別常用的方法

- setTitle(String title)
 - ◆ 設定視窗標題列的文字。
 - ◆ 若省略本行，所建立的視窗沒有標題文字。

JFrame

- `setIconImage(Image img)`
 - ◆ 設定視窗標題的圖示。
 - ◆ 若省略，則使用預設圖示。

注意：

當方法中要傳遞Image物件時，須先引入**`import java.awt.*`**套件，再透過Image類別的方法來建立Image物件。

例如：

Image 物件名稱 = Toolkit.getDefaultToolkit().getImage("Image的路徑");

- `setSize(int width, int height)`
 - ◆ 設定視窗的寬度與高度（即視窗大小），單位為像素（pixel）。
- `setLocation(int x, int y)`
 - ◆ 設定視窗左上角的（x, y）座標（即視窗位置），單位為像素（pixel）。

JFrame

- `setBounds(int x, int y, int width, int height)`
 - ◆ 設定視窗的左上角的(x, y)坐標與視窗的寬度與高度。
- `setVisible(boolean b)`
 - ◆ 設定視窗是否顯示。
 - ◆ 若引數b為true時，表示視窗可顯示；
 - ◆ 若引數b為false時，表示視窗隱藏。
- `setResizable(boolean b)`
 - ◆ 設定此視窗是否可以調整大小，預設值為true。
 - ◆ 當b為false時，視窗無法調整。
- `setLayout(LayoutManager manager)`
 - ◆ 指定視窗的版面配置方式。
 - ◆ 如果不使用系統提供的版面配置則呼叫本方法時，引數要設為null，即「**`setLayout(null);`**」，此時視窗內的每一個swing元件，需要設定在視窗內的位置坐標。

JFrame

- `setDefaultCloseOperation(int op)`
 - ◆ 當視窗最上面標題列右側的關閉鈕被點按時，系統會採取的動作。
 - ◆ 引數op共有四種設定：
 - ◆ `JFrame.DO_NOTHING_ON_CLOSE`
不採取任何動作。
 - ◆ `JFrame.HIDE_ON_CLOSE`
預設值，隱藏視窗。
 - ◆ `JFrame.DISPOSE_ON_CLOSE`
移除視窗。
 - ◆ `JFrame.EXIT_ON_CLOSE`
移除視窗並結束應用程式。

注意：

- ◆ 當視窗隱藏或移除，但應用程式並未結束時，在主控制台模式可以按Ctrl+C鍵，強迫結束應用程式。

JFrame

- add(Component comp)
 - ◆ comp是指按鈕（JButton）、標籤（JLabel）、文字欄位（JTextField）...等swing元件。
 - ◆ 當這些元件要放入視窗時，要使用add()方法。
- pack()
 - ◆ 依據視窗內容，自動調整視窗大小。
 - ◆ 不可與setSize()同時使用。
 - ◆ 若與setSize()同時使用，則pack()會失效。
- repaint()
 - ◆ 重新繪製元件。

JFrame

◆ 程式：

```
package CH09_01;
```

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

```
public class CH09_01
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        JFrame frame=new JFrame("Swing 視窗程式");
```

```
        frame.setVisible(true);
```

```
        frame.setBounds(150,100,400,200);
```

```
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
    }
```

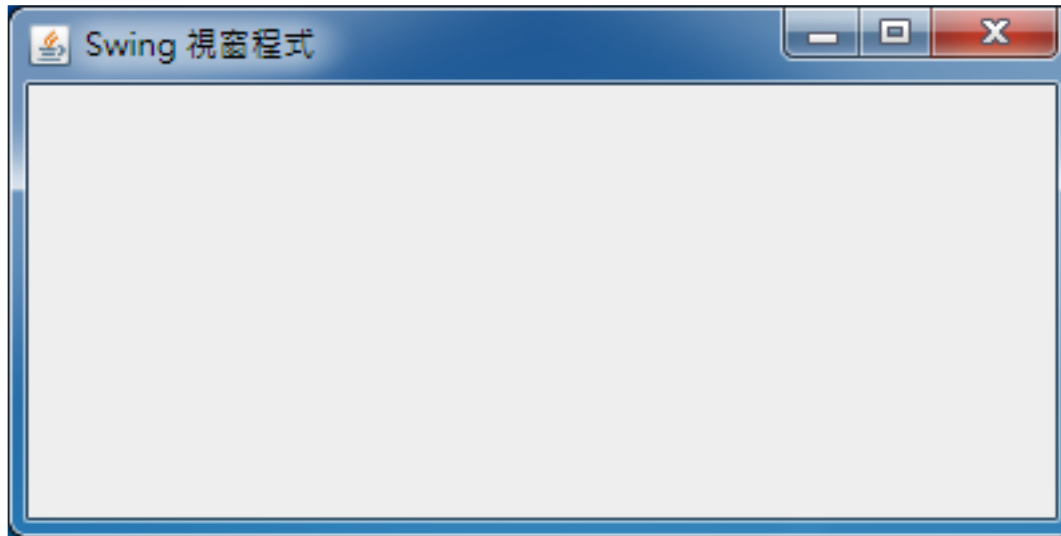
```
}
```

JFrame

```
1 package CH09_01;
2
3 import javax.swing.*;
4
5 public class CH09_01
6 {
7     public static void main(String[] args)
8     {
9         JFrame frame=new JFrame("Swing 視窗程式");
10        frame.setVisible(true);
11        frame.setBounds(150,100,400,200);
12        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13    }
14 }
```

JFrame

◆ 執行結果：



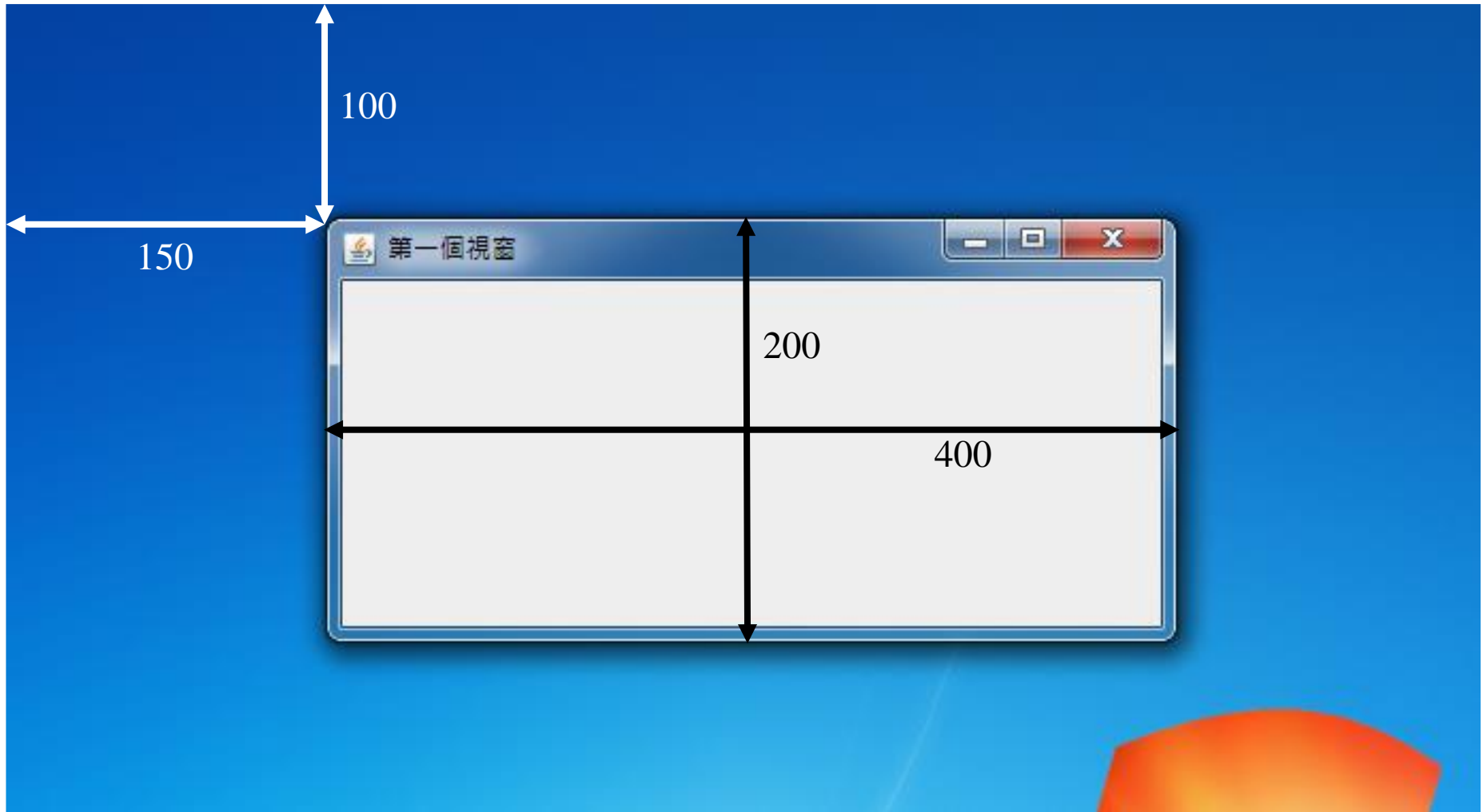
◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。

JFrame

- 行03：
 - ◆ 載入「javax.swing.*」套件。
- 行09：
 - ◆ 利用「JFrame(String title)」的「建構子」，建立frame物件。
- 行10：
 - ◆ 設定視窗可以顯示在螢幕上。
- 行11：
 - ◆ 設定視窗的左上角與螢幕左上方的距離與視窗大小。
 - ◆ 此例設定視窗「左上角」位置距離螢幕左方150像素、上方100像素；視窗大小為寬400像素（包含「左、右緣框線」各「8像素」）、高200像素（包含「標題列」「30像素」及「下緣框線」「8像素」）（如下圖）。

JFrame



視窗「高度」包含標題列30像素及下緣框線8像素，
視窗「寬度」包含左、右緣框線各「8像素」。

JFrame

- 行12：
 - ◆ 設定當點按視窗右上角「關閉鈕」關閉視窗時一併「結束」應用程式。

注意：

- 行10~行12的順序互換，不影響程式正常執行。
- 若省略行12的敘述，則關閉視窗時，「應用程式仍在背景執行，尚未結束」，若要關閉該視窗（即結束應用程式），
 - ◆ 若在主控制台模式下，按「Ctrl+ C」鍵，強迫結束應用程式。
 - ◆ 若在IDE（整合開發環境）模式下，建議使用Windows的工作管理員來結束應用程式。

JFrame

◆ 程式：

```
package CH09_02;
```

```
import java.awt.*;
```

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

```
class CMyFrame extends JFrame
```

```
{
```

```
    CMyFrame()
```

```
{
```

```
    Image img = Toolkit.getDefaultToolkit().getImage(  
        "../CH09/src/CH09_02/images/octopus.gif");
```

```
    setTitle("Swing 視窗程式 2");
```

```
    setIconImage(img);
```

```
    setLocation(550, 300);
```

```
    setSize(400, 200);
```

```
    setVisible(true);
```

```
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

JFrame

```
}  
}
```

```
public class CH09_02
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        JFrame frame1 = new JFrame("Swing 視窗程式 1");
```

```
        frame1.setVisible(true);
```

```
        frame1.setBounds(150, 100, 400, 200);
```

```
        frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        CMyFrame frame2 = new CMyFrame();
```

```
    }
```

```
}
```

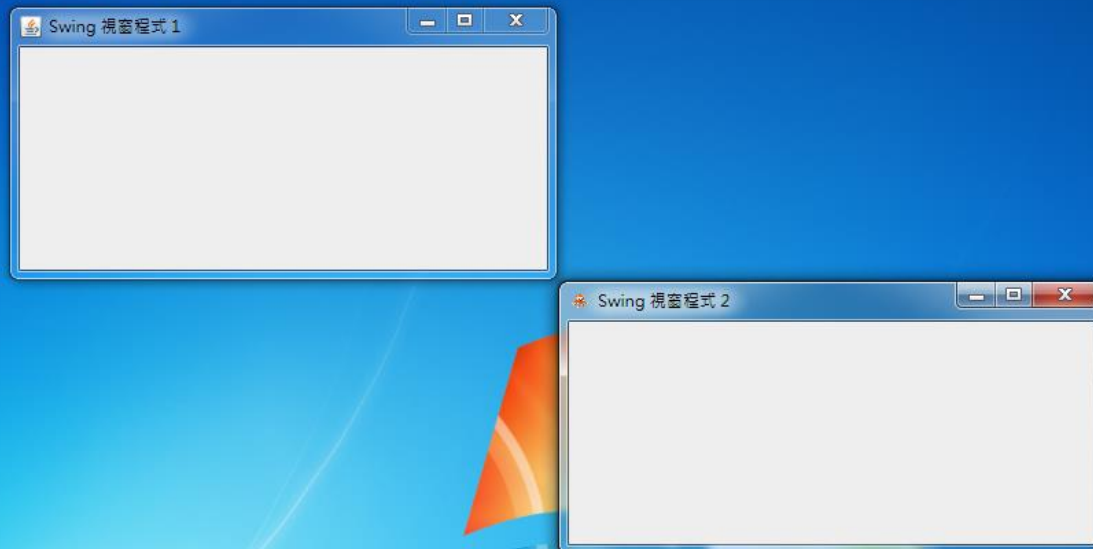
JFrame

```
1 package CH09_02;
2
3 import java.awt.*;
4 import javax.swing.*;
5
6 class CMyFrame extends JFrame
7 {
8     CMyFrame()
9     {
10         Image img = Toolkit.getDefaultToolkit().getImage("../CH09/src/CH09_02/images/octopus.gif");
11
12         setTitle("Swing 視窗程式 2");
13         setIconImage(img);
14         setLocation(550, 300);
15         setSize(400, 200);
16         setVisible(true);
17         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18     }
19 }
20
21 public class CH09_02
22 {
23     public static void main(String[] args)
24     {
25         JFrame frame1 = new JFrame("Swing 視窗程式 1");
26         frame1.setVisible(true);
```

JFrame

```
27 frame1.setBounds(150, 100, 400, 200);
28 frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
29
30 CMyFrame frame2 = new CMyFrame();
31 }
32 }
```

◆ 執行結果：



JFrame

◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。
- 行03：
 - ◆ 載入「java.awt.*」套件。
- 行04：
 - ◆ 載入「javax.swing.*」套件。
- 行06~行19：
 - ◆ 建立繼承自「JFrame」父類別的「CMyFrame」子類別。
 - ◆ 行08~行18：
 - ◆ 宣告類別「建構子」 「CMyFrame()」。
 - ◆ 行10：
 - 建立Image物件img。

JFrame

- ◆ 行12：
設定視窗標題列的文字。
- ◆ 行13：
設定視窗圖示。
- ◆ 行14：
設定視窗的位置。
- ◆ 行15：
設定視窗的大小。
- ◆ 行16：
設定視窗可以顯示在螢幕上。
- ◆ 行17：
設定當點按視窗右上角關閉鈕時，一併結束應用程式。

JFrame

- 行25：
 - ◆ 利用「JFrame(String title)」的「建構子」，建立frame1物件。
- 行26：
 - ◆ 設定視窗可以顯示在螢幕上。
- 行27：
 - ◆ 設定視窗的位置與視窗的大小。
- 行28：
 - ◆ 設定當點按視窗右上角關閉鈕時，一併結束應用程式。
- 行30：
 - ◆ 利用「CMyFrame()」的「建構子」，建立frame2物件。

Swing元件常用的方法

下面列出一些Swing元件常用的方法，Swing套件內的元件大多可以使用，但會有少許方法不適用於特定元件，需要時再進一步說明。

◆ setText(String text)

- 設定元件內的顯示文字。

◆ getText()

- 傳回元件內的文字字串。

◆ setBounds(int x, int y, int w, int h)

- 設定元件的左上角坐標（即元件的位置）與元件的大小。
- 坐標位置參考原點在視窗內部的左上角，單位為像素（pixel）。

Swing元件常用的方法

- ◆ setLocation(int x, int y)
 - 設定元件的左上角(x, y)坐標。
- ◆ getX()
 - 傳回元件的x坐標。
- ◆ getY()
 - 傳回元件的y坐標。
- ◆ setSize(int w, int h)
 - 設定元件的寬度及高度（即 元件的大小）。
- ◆ getHeight()
 - 傳回元件的高度。
- ◆ getWidth()
 - 傳回元件的寬度。

Swing元件常用的方法

◆ setVisible(boolean b)

- 設定元件是否顯示。

◆ isVisible()

- 判斷元件是否顯示。
- 若顯示元件則傳回true，否則傳回false。

◆ setEnabled(Boolean b)

- 設定元件是否有作用（是否可以使用）。

◆ isEnabled()

- 判斷元件是否有作用（是否可以使用）。

◆ setIcon(ImageIcon icon)

- 設定元件內的顯示圖形物件。

Swing元件常用的方法

注意：

因為Icon是一個介面，沒有建構式，所以當方法中要傳遞Icon物件時，可以透過ImageIcon類別來建立物件，例如：

ImageIcon icon = new ImageIcon("Icon的路徑");

◆ getIcon()

- 傳回元件內的圖形物件。

◆ setFont(Font f)

- 設定物件的字型。

注意：

當方法中要傳遞Font物件時，須先引入

import java.awt.Font套件，再透過Font類別的方法來建立Font物件。

Swing元件常用的方法

例如：

Font 物件名稱=new Font(String name, int style, int size);

- ◆ name：

- ◆ 字體名稱，必須是Java提供的字型。

- ◆ style：

- ◆ 字型樣式。

- ◆ 利用Font常數來設定，有下列幾種樣式：

- Font.BOLD → 粗體

- Font.ITALIC → 斜體

- Font.PLAIN → 標準

- ◆ 如果要同時設定兩種樣式，可以將Font常數相加，例如：Font f=new Font("Arial", Font.BOLD +Font.ITALIC, 20);，表示粗斜體。

- ◆ size：

- ◆ 字體大小。

JLabel

- 標籤在視窗應用程式界面中，常被用來提示操作訊息或做為某項動作的標題文字。
- 標籤物件的建立要使用Swing套件的JLabel類別來宣告，而且要使用JFrame類別的add()方法，才能將標籤物件放入視窗物件中。
- 標籤的呈現樣式，可以只有文字訊息或只顯示圖片，也可以在文字前面顯示圖片。
- 當用JLabel類別來宣告建立標籤物件時，同樣會執行JLabel類別的建構子。

JLabel

◆ JLabel類別的建構子：

- JLabel()
 - ◆ 建立標籤物件，標籤上沒有顯示文字，也沒有圖片。
- JLabel(String text)
 - ◆ 建立標籤物件，標籤上顯示text文字字串。
- JLabel(ImageIcon icon)
 - ◆ 建立標籤物件，標籤上顯示圖片。
- JLabel(String text, int align)
 - ◆ 引數String text為標籤文字。
 - ◆ 引數align是標籤文字在物件本身寬度範圍內的水平對齊方式，有：
 - ◆ SwingConstants.LEFT
預設值，靠左對齊。

JLabel

- ◆ `SwingConstants.RIGHT`
靠右對齊。
- ◆ `SwingConstants.CENTER`
置中對齊。
- `JLabel(String text, ImageIcon icon, int align)`
 - ◆ 建立有文字、圖片的標籤物件，圖片會緊接在文字的前面。

◆ JLabel類別常用的方法：

- `setIcon(Icon icon)`
 - ◆ 設定標籤所顯示的圖示。
- `getIcon()`
 - ◆ 取得標籤所顯示的圖示。

JLabel

◆ 程式：

```
package CH09_03;

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

public class CH09_03
{
    public static void main(String[] args)
    {
        JFrame frame = new JFrame("JLabel");
        frame.setLayout(null);
        frame.setBounds(100, 100, 300, 150);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Font f=new Font("標楷體", Font.BOLD, 12);
        ImageIcon icon = new ImageIcon(
            "..\\CH09\\src\\CH09_03\\images\\folder.gif");
    }
}
```

JLabel

```
JLabel lbl1 = new JLabel("1、標籤文字");
lbl1.setFont(f);
lbl1.setBounds(0, 0, 80, 12);
frame.add(lbl1);

JLabel lbl2 = new JLabel("2、圖示及標籤文字", icon,
                        SwingConstants.RIGHT);
lbl2.setBounds(20, 20, 260, 30);
frame.add(lbl2);

frame.repaint();
}
}
```

JLabel

```
1 package CH09_03;
2
3 import java.awt.*;
4 import javax.swing.*;
5
6 public class CH09_03
7 {
8     public static void main(String[] args)
9     {
10         JFrame frame = new JFrame("JLabel");
11         frame.setLayout(null);
12         frame.setBounds(100, 100, 300, 150);
13         frame.setVisible(true);
14         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15
16         Font f=new Font("標楷體", Font.BOLD, 12);
17         ImageIcon icon = new ImageIcon("../CH09/src/CH09_03/images/folder.gif");
18
19         JLabel lbl1 = new JLabel("1、標籤文字");
20         lbl1.setFont(f);
21         lbl1.setBounds(0, 0, 80, 12);
22         frame.add(lbl1);
23
24         JLabel lbl2 = new JLabel("2、圖示及標籤文字", icon, SwingConstants.RIGHT);
25         lbl2.setBounds(20, 20, 260, 30);
26         frame.add(lbl2);
```

JLabel

```
27  
28     frame.repaint();  
29 }  
30 }
```

◆ 執行結果：



JLabel

◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。
- 行03：
 - ◆ 載入「java.awt.*」套件。
- 行04：
 - ◆ 載入「javax.swing.*」套件。
- 行10：
 - ◆ 利用「JFrame(String title)」的「建構子」，建立frame物件。
- 行11：
 - ◆ 不使用系統提供的版面配置。

JLabel

- 行12：
 - ◆ 設定視窗的位置與視窗的大小。
- 行13：
 - ◆ 設定視窗可以顯示在螢幕上。
- 行14：
 - ◆ 設定當點按視窗右上角關閉鈕時，一併結束應用程式。
- 行16：
 - ◆ 建立字型物件f。
- 行17：
 - ◆ 利用「ImageIcon(String filename)」的「介面」，建立icon物件。
- 行19：
 - ◆ 利用「JLabel(String text)」的「建構子」，建立lbl1物件。

JLabel

- 行20：
 - ◆ 設定字型。
- 行21：
 - ◆ 設定lbl1物件的位置與大小。
- 行22：
 - ◆ 將lbl1物件，增加到frame物件中。
- 行24：
 - ◆ 利用「JLabel(String text, Icon icon, int horizontalAlignment)」的「建構子」，建立lbl2物件。
- 行25：
 - ◆ 設定lbl2物件的位置與大小。
- 行26：
 - ◆ 將lbl2物件，增加到frame物件中。

JLabel

- 行28：
 - ◆ 重繪視窗元件。

JTextField

■ JTextField（單行文字欄位）類別最主要的功能是可以讓使用者透過鍵盤來輸入資料或修改資料或顯示文字訊息。

◆ JTextField類別的建構子：

- JTextField()
 - ◆ 建立空白的單行文字欄位物件。
- JTextField(String text)
 - ◆ 建立並顯示text文字字串的單行文字欄位物件。

◆ JTextField類別常用的方法：

- setText(String text)
 - ◆ 設定文字欄位所要顯示的文字。
- getText()
 - ◆ 取得文字欄位所顯示的文字。

JTextField

- `setEditable(boolean b)`
 - ◆ 設定文字欄位是否可輸入資料。
 - ◆ 若引數b為true時，表示可輸入資料；
 - ◆ 若引數b為false時，表示文字欄位為唯讀狀態（不能鍵入資料）。

JTextField

◆ 程式：

```
package CH09_04;

import javax.swing.*;

public class CH09_04
{
    public static void main(String[] args)
    {
        JFrame frame = new JFrame("JTextField");
        frame.setLayout(null);
        frame.setBounds(100, 100, 300, 150);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JTextField txt = new JTextField("單行文字欄位");
        txt.setBounds(0, 50, 120, 20);
        txt.setEditable(false);
        frame.add(txt);
    }
}
```

JTextField

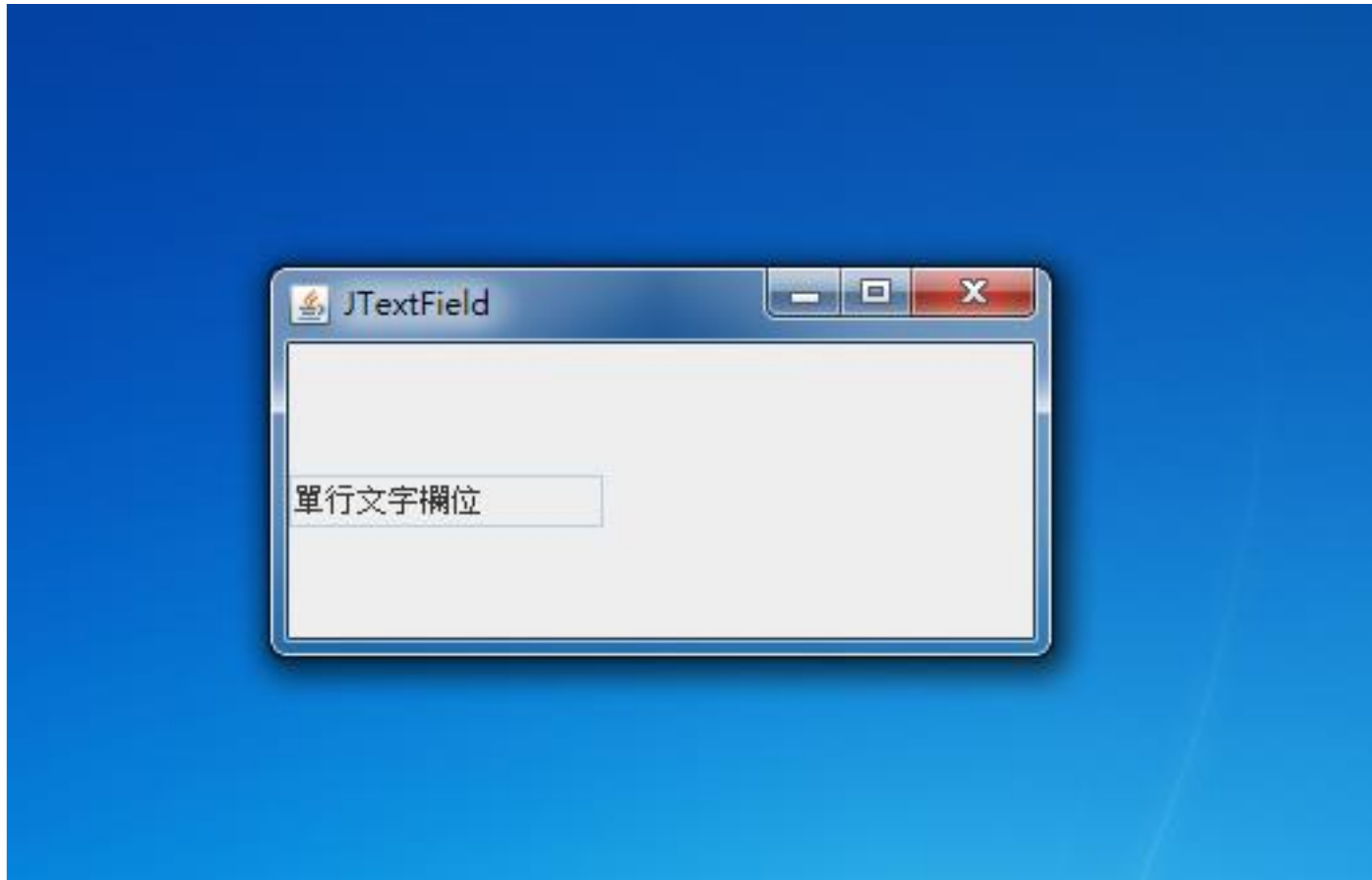
```
        frame.repaint();  
    }  
}
```

JTextField

```
1 package CH09_04;
2
3 import javax.swing.*;
4
5 public class CH09_04
6 {
7     public static void main(String[] args)
8     {
9         JFrame frame = new JFrame("JTextField");
10        frame.setLayout(null);
11        frame.setBounds(100, 100, 300, 150);
12        frame.setVisible(true);
13        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
14
15        JTextField txt = new JTextField("單行文字欄位");
16        txt.setBounds(0, 50, 120, 20);
17        txt.setEditable(false);
18        frame.add(txt);
19
20        frame.repaint();
21    }
22 }
```

JTextField

◆ 執行結果：



JTextField

◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。
- 行03：
 - ◆ 載入「javax.swing.*」套件。
- 行09：
 - ◆ 利用「JFrame(String title)」的「建構子」，建立frame物件。
- 行10：
 - ◆ 不使用系統提供的版面配置。
- 行11：
 - ◆ 設定視窗的位置與視窗的大小。

JTextField

- 行12：
 - ◆ 設定視窗可以顯示在螢幕上。
- 行13：
 - ◆ 設定當點按視窗右上角關閉鈕時，一併結束應用程式。
- 行15：
 - ◆ 利用「JTextField(String text)」的「建構子」，建立txt物件。
- 行16：
 - ◆ 設定txt物件的位置與大小。
- 行17：
 - ◆ 設定txt物件為唯讀（即不可輸入資料）。
- 行18：
 - ◆ 將txt物件，增加到frame物件中。

JTextField

- 行20：
 - ◆ 重繪視窗元件。

JPasswordField

■ JPasswordField（密碼文字欄位）類別是當使用者透過鍵盤來輸入資料時，以特定字元顯示。

◆ JPasswordField類別的建構子：

- JPasswordField()
 - ◆ 建立密碼文字欄位物件。
- JPasswordField(String text)
 - ◆ 建立含預設文字的密碼文字欄位。

◆ JPasswordField類別常用的方法：

- setEchoChar(Char char)
 - ◆ 所輸入的字元以char指定的字元顯示。
 - ◆ 省略以「.」字元顯示。

JPasswordField

- getPassword()
 - ◆ 取得密碼文字欄位內所輸入的文字。
 - ◆ 所取得文字需轉為字串物件才能做字串資料的處理。

例如：

```
String pw = new String(密碼文字欄位物件.getPassword());
```

JPasswordField

◆ 程式：

```
package CH09_05;
```

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

```
public class CH09_05  
{
```

```
    public static void main(String[] args)  
    {
```

```
        JFrame frame = new JFrame("JTextField");  
        frame.setLayout(null);  
        frame.setBounds(100, 100, 300, 150);  
        frame.setVisible(true);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        JPasswordField pws = new JPasswordField("password");  
        pws.setBounds(0, 50, 120, 20);  
        pws.setEchoChar('*');  
        frame.add(pws);
```

JPasswordField

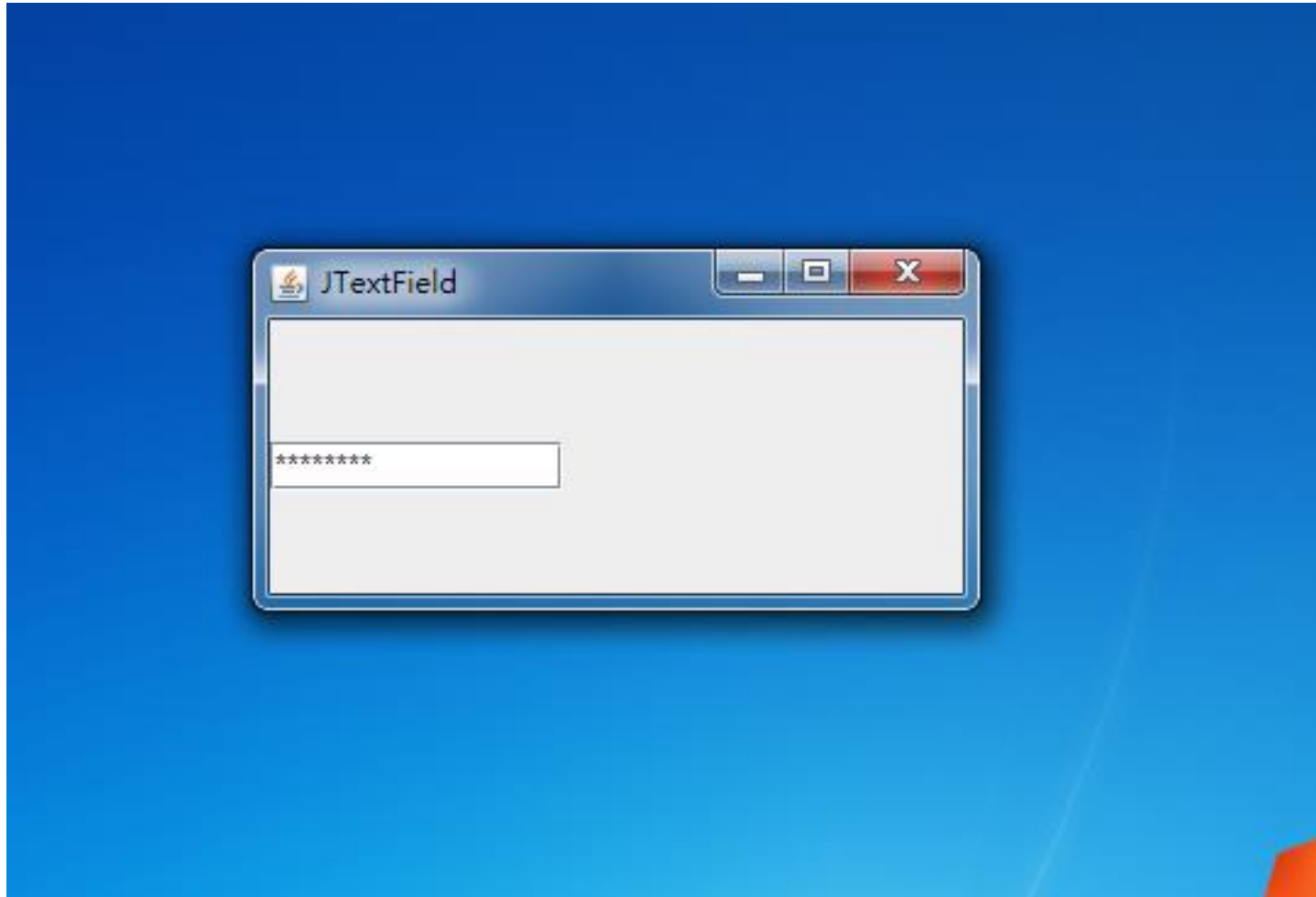
```
        frame.repaint();  
    }  
}
```

JPasswordField

```
1 package CH09_05;
2
3 import javax.swing.*;
4
5 public class CH09_05
6 {
7     public static void main(String[] args)
8     {
9         JFrame frame = new JFrame("JTextField");
10        frame.setLayout(null);
11        frame.setBounds(100, 100, 300, 150);
12        frame.setVisible(true);
13        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
14
15        JPasswordField pws = new JPasswordField("password");
16        pws.setBounds(0, 50, 120, 20);
17        pws.setEchoChar('*');
18        frame.add(pws);
19
20        frame.repaint();
21    }
22 }
```

JPasswordField

◆ 執行結果：



JPasswordField

◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。
- 行03：
 - ◆ 載入「javax.swing.*」套件。
- 行09：
 - ◆ 利用「JFrame(String title)」的「建構子」，建立frame物件。
- 行10：
 - ◆ 不使用系統提供的版面配置。
- 行11：
 - ◆ 設定視窗的位置與視窗的大小。

JPasswordField

- 行12：
 - ◆ 設定視窗可以顯示在螢幕上。
- 行13：
 - ◆ 設定當點按視窗右上角關閉鈕時，一併結束應用程式。
- 行15：
 - ◆ 利用「JPasswordField(String text)」的「建構子」，建立pws物件。
- 行16：
 - ◆ 設定pws物件的位置與大小。
- 行17：
 - ◆ 設定在pws物件中所輸入的字元以「*」顯示。
- 行18：
 - ◆ 將pws物件，增加到frame物件中。

JPasswordField

- 行20：
 - ◆ 重繪視窗元件。

JButton

- 按鈕（Button）在視窗程式界面中的主要功能是確認所要執行的動作，其使用的頻率非常的高。
- 在Java中，可以使用JButton類別來宣告建立按鈕物件。
- 按鈕上可以顯示文字、圖片或兩者兼具。
 - ◆ JButton類別的建構子如下：
 - JButton()
 - ◆ 建立按鈕物件，按鈕上沒有顯示文字，也沒有圖片。
 - JButton(String text)
 - ◆ 立的按鈕物件，並顯示text文字字串。
 - JButton(ImageIcon icon)
 - ◆ 建立的按鈕物件，並顯示圖片物件。

JButton

- JButton(String text, ImageIcon icon)
 - ◆ 建立的按鈕物件，顯示文字與圖片。

JButton

◆ 程式：

```
package CH09_06;
```

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

```
public class CH09_06  
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        JFrame frame = new JFrame("JTextField");
```

```
        frame.setLayout(null);
```

```
        frame.setBounds(100, 100, 300, 150);
```

```
        frame.setVisible(true);
```

```
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        ImageIcon icon = new ImageIcon(  
            "..\\CH09\\src\\CH09_06\\images\\Horse.gif");
```

```
        JButton btn = new JButton("按鈕", icon);
```

JButton

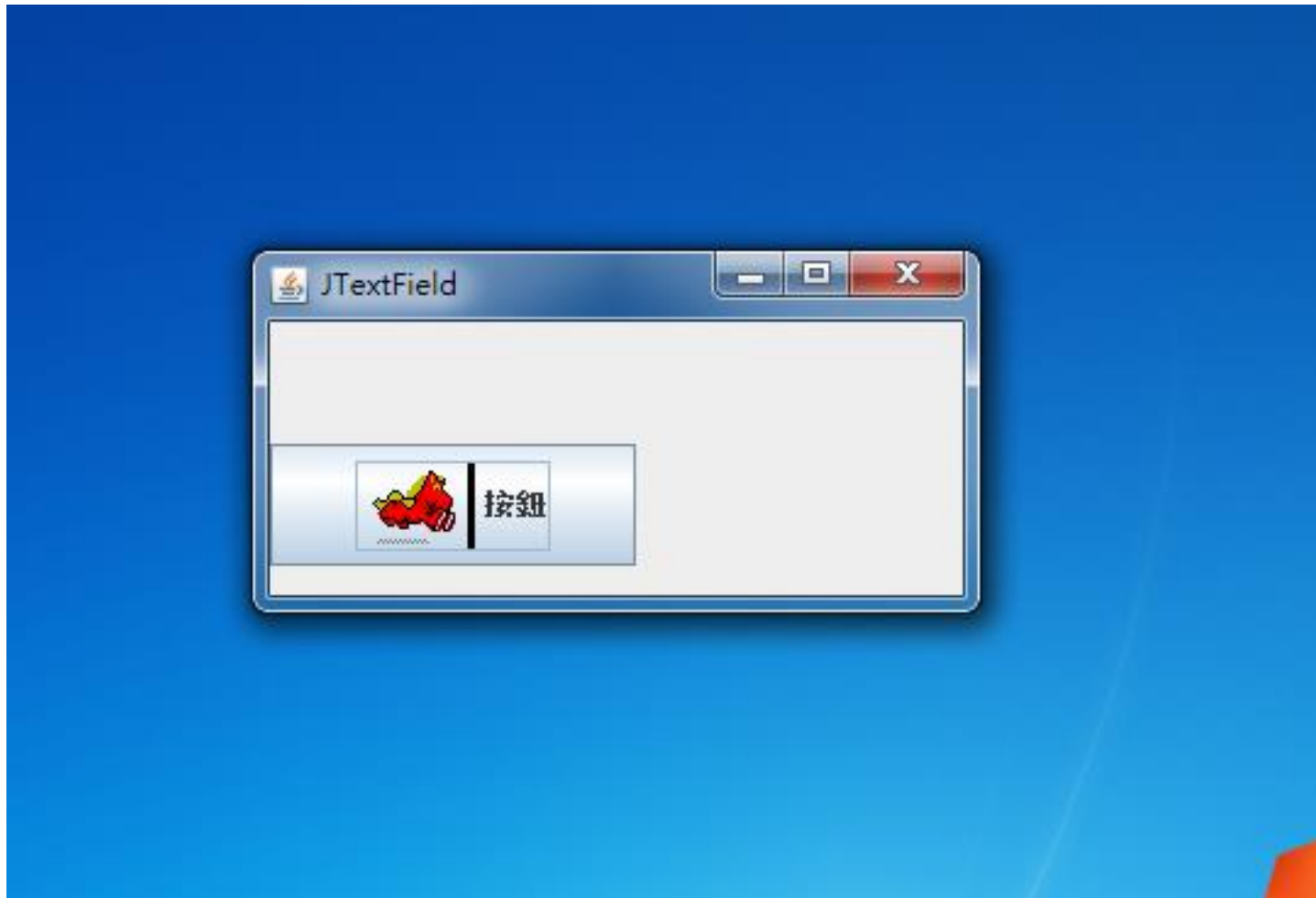
```
        btn.setBounds(0, 50, 150, 50);  
        frame.add(btn);  
  
        frame.repaint();  
    }  
}
```

JButton

```
1 package CH09_06;
2
3 import javax.swing.*;
4
5 public class CH09_06
6 {
7     public static void main(String[] args)
8     {
9         JFrame frame = new JFrame("JTextField");
10        frame.setLayout(null);
11        frame.setBounds(100, 100, 300, 150);
12        frame.setVisible(true);
13        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
14
15        ImageIcon icon = new ImageIcon("../CH09/src/CH09_06/images/Horse.gif");
16
17        JButton btn = new JButton("按鈕", icon);
18        btn.setBounds(0, 50, 150, 50);
19        frame.add(btn);
20
21        frame.repaint();
22    }
23 }
```

JButton

◆ 執行結果：



JButton

◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。
- 行03：
 - ◆ 載入「javax.swing.*」套件。
- 行09：
 - ◆ 利用「JFrame(String title)」的「建構子」，建立frame物件。
- 行10：
 - ◆ 不使用系統提供的版面配置。
- 行11：
 - ◆ 設定視窗的位置與視窗的大小。

JButton

- 行12：
 - ◆ 設定視窗可以顯示在螢幕上。
- 行13：
 - ◆ 設定當點按視窗右上角關閉鈕時，一併結束應用程式。
- 行15：
 - ◆ 利用「`ImageIcon(String filename)`」的「建構子」，建立icon物件。
- 行17：
 - ◆ 利用「`JButton(String text, Icon icon)`」的「建構子」，建立btn物件。
- 行18：
 - ◆ 設定btn物件的位置與大小。
- 行19：
 - ◆ 將btn物件，增加到frame物件中。

JButton

- 行21：
 - ◆ 重繪視窗元件。

綜合練習

題目：建立一視窗

要求如下圖：



登入

請輸入帳號及密碼

帳號：

密碼：

確定 重來

資料來源

- 蔡文龍、何嘉益、張志成、張力元，JAVA SE 10基礎必修課，台北市，碁峰資訊股份有限公司，2018年7月，出版。
- 吳燦銘、胡昭民，圖解資料結構-使用Java(第三版)，新北市，博碩文化股份有限公司，2018年5月，出版。
- Ivor Horton，Java 8 教學手冊，台北市，碁峰資訊股份有限公司，2016年9月，出版。
- 李春雄，程式邏輯訓練入門與運用---使用JAVA SE 8，台北市，上奇科技股份有限公司，2016年6月，初版。
- 位元文化，Java 8視窗程式設計，台北市，松崗資產管理股份有限公司，2015年12月，出版。
- Benjamin J Evans、David Flanagan，Java 技術手冊 第六版，台北市，碁峰資訊股份有限公司，2015年7月，出版。
- 蔡文龍、張志成，JAVA SE 8 基礎必修課，台北市，碁峰資訊股份有限公司，2014年11月，出版。
- 陳德來，Java SE 8程式設計實例，台北市，上奇科技股份有限公司，2014年11月，初版。
- 林信良，Java SE 8 技術手冊，台北市，碁峰資訊股份有限公司，2014年6月，出版。
- 何嘉益、黃世陽、李篤易、張世杰、黃鳳梅，徐政棠譯，JAVA2 程式設計從零開始--適用JDK7，台北市，上奇資訊股份有限公司，2012年5月，出版。