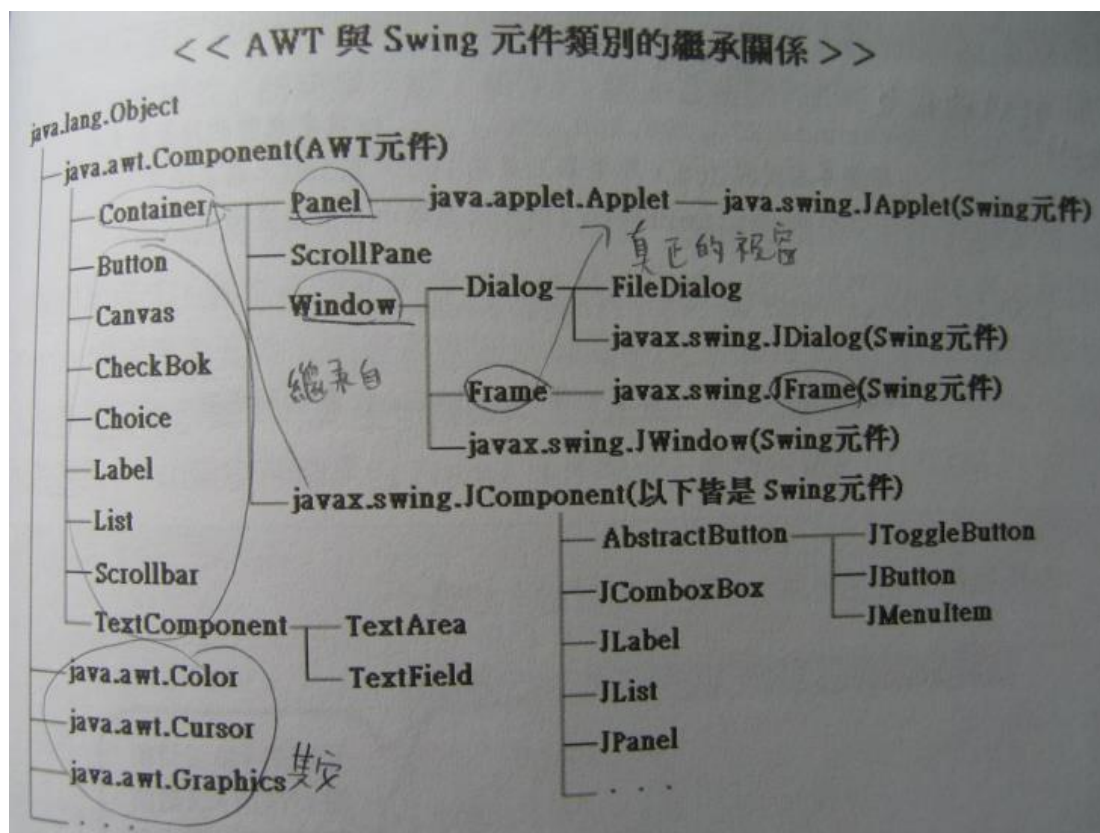


Java GUI 介面設計 → AWT

1. java 的內建類別提供了整套的圖形化元件讓我們使用
2. JDK1.0 提供了 AWT 在 `java.awt` package 類別庫
3. JDK1.1 又新增了 Swing 在 `javax.swing` package 類別庫
4. 兩個系列的元件有許多功能非常相似，不過 Swing 系列的元件類別名稱，開頭通常會多一個 J 字母
5. 例如 AWT 的視窗框架是 `Frame` 類別，而 Swing 的則是 `JFrame` 類別，又 AWT 有 `Label` 類別，而 Swing 有 `JLabel` 類別
6. 以 AWT 這一系列的元件，主要可分為兩大類
 - <1>. 可以盛裝元件的“收納器”(Container) → 重型元件
 - <2>. 另一種則是無法盛裝元件的“一般元件” → 輕型元件
7. 但“收納器”中只有一些可直接獨立浮動在螢幕上，例如 `Frame`，其它不能獨立浮動在螢幕上的收納器和一般元件，就只能依附在視窗框架上，藉此才能顯示在螢幕上
8. Swing 系列的元件由於它基礎的 `JComponent` 元件是繼承自 AWT 之 `Container` 元件而 Swing 系列常用的圖形化元件幾乎都是繼承自 `JComponent` 元件類別，因此 Swing 系列的元件幾乎都能夠盛裝元件，也較為精美



java 視窗應用程式的程式碼架構

1. 雖然繼承自 Container 的元件類別有很多，但一般用來作為視窗基本框架的，通常是指能夠直接獨立顯示在螢幕上並且可在螢幕上浮動的 Frame 或 JFrame 元件
2. 由於 Frame 或 JFrame 元件只是一個空白的視窗框架，而我們所要開發的視窗應用程式，通常在視窗內會需要許多的元件
3. 因此我們通常不是直接使用 Frame 類別產生元件，而是要宣告一個繼承自 Frame 的類別，並在裡面加入元件作為它的成員，然後再利用該類別去產生一個 “擁有所需之元件內容與功能的視窗”

此類別若是繼承自 Frame 元件類別，則程式碼架構

```
public class Frame0101Frame表單架構 extends Frame {
    public TextField T1;
    public Button B1;
    public Label L1;
    public Frame0101Frame表單架構() {
        //手動指定各元件的位置 Frame 的setLayout 方式
        this.setLayout(null);
        //元件產生區
        T1 = new TextField("我是文字方塊"); //1
        T1.setBounds(100, 60, 90, 28); //2
        this.add(T1); //將 T1 加到 Frame 內 //4

        B1 = new Button("我是按鈕"); //1
        B1.setBounds(100, 110, 90, 28); //2
        B1.addMouseListener(new myMouseListener()); //3
        this.add(B1); //將 B1 加到 Frame 內 //4

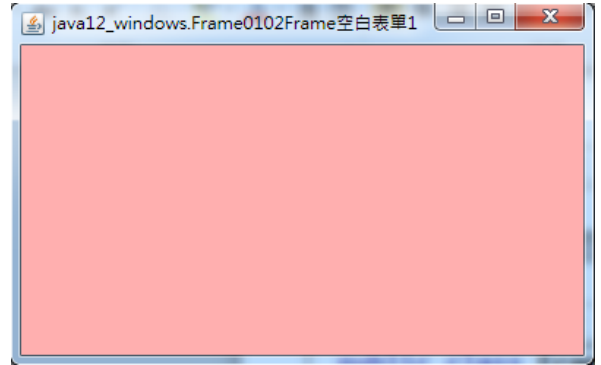
        L1 = new Label("我是標籤"); //1
        L1.setBounds(100, 140, 90, 28); //2
        this.add(L1); //將 L1 加到 Frame 內 //4
        //////////////////////////////////////
        this.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 300, 250);
        this.setVisible(true);
    }
    public class myMouseListener implements MouseListener {

        public void mouseClicked(MouseEvent e) {
        }
        public void mouseEntered(MouseEvent e) {
        }
        public void mouseExited(MouseEvent e) {
        }
        public void mousePressed(MouseEvent e) {
        }
        public void mouseReleased(MouseEvent e) {
        }
    }
    public static void main(String[] args) {
        new Frame0101Frame表單架構();
    }
}
```

```
public class Frame0102Frame空白表單1 extends Frame {

    public Frame0102Frame空白表單1() {
        this.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        //距螢幕左邊緣200，距螢幕上邊緣100，寬300，高250。
        this.setBounds(200, 100, 400, 250);
        //令此視窗顯示在螢幕畫面上
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0102Frame空白表單1();
    }
}
```



```
public class Frame0102Frame空白表單2 {

    public static void main(String[] args) {
        Frame cp = new Frame();

        cp.setBackground(Color.PINK);
        cp.setTitle("空白表單");
        cp.setBounds(200, 100, 400, 250);
        cp.setVisible(true);
    }
}
```

```
public class Frame0103Frame表單元件1 extends Frame {
```

```
    public TextField T1;
    public Button B1;

    public Frame0103Frame表單元件1() {
        this.setLayout(null);

        T1 = new TextField("我愛你"); //1
        T1.setBounds(100, 60, 90, 28); //2
        // T1.setLocation(100, 60);
        // T1.setSize(90, 28);
        this.add(T1); //將 T1 加到 Frame 內 //4

        B1 = new Button("Button"); //1
        B1.setBounds(100, 110, 90, 28); //2
        // B1.setLocation(100, 110);
        // B1.setSize(90, 28);
        this.add(B1); //將 B1 加到 Frame 內 //4

        this.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 400, 250);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0103Frame表單元件1();
    }
}
```



```
public class Frame0103Frame表單元件2 extends Frame {

    public TextField T1;
    public Button B1;

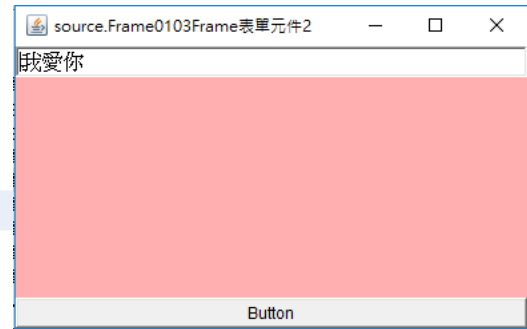
    public Frame0103Frame表單元件2() {
        BorderLayout layout = new BorderLayout();
        this.setLayout(layout);

        T1 = new TextField("我愛你"); //1
        // T1.setBounds(100, 60, 90, 28); //2
        this.add(T1, BorderLayout.NORTH); //將 T1 加到 Frame 內 //4

        B1 = new Button("Button"); //1
        // B1.setBounds(100, 110, 90, 28); //2
        this.add(B1, BorderLayout.SOUTH); //將 B1 加到 Frame 內 //4

        this.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 400, 250);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0103Frame表單元件2();
    }
}
```



此類別若是繼承自 JFrame 元件類別，則程式碼架構

```

public class Frame0104JFrame表單架構 extends JFrame {
    public JTextField JT1;
    public JButton JB1;
    public JLabel JL1;
    public Container ContentPane;
    public Frame0104JFrame表單架構() {
        //取回內容桌布
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);

        JT1 = new JTextField("我是文字方塊"); //1
        JT1.setBounds(100, 60, 90, 28); //2
        ContentPane.add(JT1); //將 JT1 加到 內容桌布內 //4

        JB1 = new JButton("我是按鈕"); //1
        JB1.setBounds(100, 110, 90, 28); //2
        JB1.addMouseListener(new myMouseListener()); //3
        ContentPane.add(JB1); //將 JB1 加到 內容桌布內 //4

        JL1 = new JLabel("我是標籤"); //1
        JL1.setBounds(100, 140, 90, 28); //2
        ContentPane.add(JL1); //將 L1 加到 內容桌布內 //4

        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 300, 250);
        this.setVisible(true);
    }
    class myMouseListener implements MouseListener {
        public void mouseClicked(MouseEvent e) {
        }
        public void mouseEntered(MouseEvent e) {
        }
        public void mouseExited(MouseEvent e) {
        }
        public void mousePressed(MouseEvent e) {
        }
        public void mouseReleased(MouseEvent e) {
        }
    }
    public static void main(String[] args) {
        new Frame0104JFrame表單架構();
    }
}

```

```
public class Frame0105JFrame表單元件1 extends JFrame {

    public JTextField JT1;
    public JButton JB1;
    public Container ContentPane;

    public Frame0105JFrame表單元件1() {

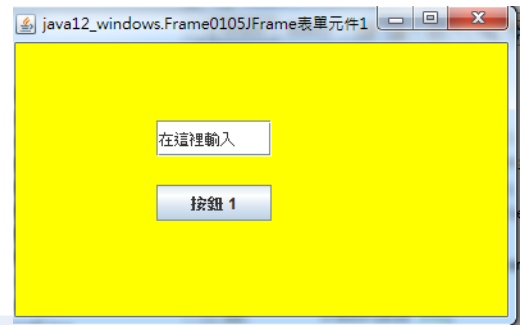
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);

        JT1 = new JTextField("在這裡輸入"); //1
        JT1.setBounds(110, 60, 90, 28); //2
        ContentPane.add(JT1); //將 JT1 加到 JFrame 的 ContentPane 內 //4

        JB1 = new JButton("按鈕 1"); //1
        JB1.setBounds(110, 110, 90, 28); //2
        ContentPane.add(JB1); //將 JB1 加到 JFrame 的 ContentPane 內 //4

        //設定 ContentPane 背景顏色為黃色
        ContentPane.setBackground(Color.yellow);
        this.setBounds(200, 100, 400, 250);
        this.setTitle(getClass().getName().toString());
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0105JFrame表單元件1();
    }
}
```



```
public class Frame0105JFrame表單元件2 extends JFrame {

    public JTextField JT1;
    public JButton JB1;
    public Container ContentPane;

    public Frame0105JFrame表單元件2() {
        initComponents();
    }

    private void initComponents() {

        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);

        JT1 = new JTextField("我愛你");
        JB1 = new JButton("我愛你");

        JT1.setBounds(100, 60, 90, 28);
        JB1.setBounds(100, 110, 90, 28);

        ContentPane.add(JT1);
        ContentPane.add(JB1);

        ContentPane.setBackground(Color.YELLOW);

        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 300, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0105JFrame表單元件2();
    }
}
```

```

public class Frame0106JFrame表單物件 {

    public static void main(String[] args) {
        JFrame cp = new JFrame();
        JTextField JT1;
        JButton JB1;

        Container ContentPane;
        ContentPane = cp.getContentPane();
        ContentPane.setLayout(null);

        JT1 = new JTextField("我愛你"); //1
        JT1.setBounds(100, 60, 90, 28); //2
        ContentPane.add(JT1); //4

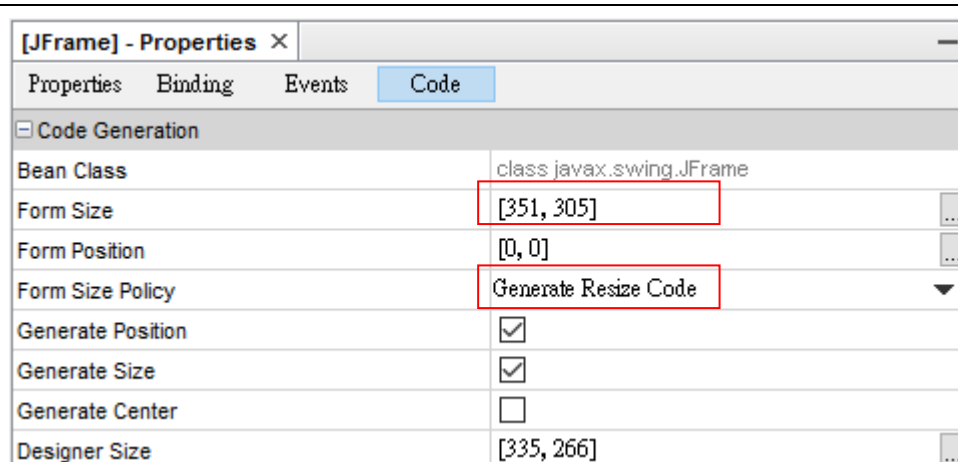
        JB1 = new JButton("Button"); //1
        JB1.setBounds(100, 110, 90, 28); //2
        ContentPane.add(JB1); //4

        ContentPane.setBackground(Color.PINK);
        cp.setTitle("表單物件");
        cp.setBounds(200, 100, 300, 250);
        cp.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        cp.setVisible(true);
    }
}

```

版面配置

1. java 設計了 Layout 這個機制來解決元件配置的問題
2. java.awt 套件中總共提供了六個基本的 Layout 類別
3. 只有繼承 Container 類別的類別才能使用 Layout
4. Java 提供的六種版面配置方式
 - <1>.BorderLayout 邊界式版面配置
 - <2>.GridLayout 方格式版面配置
 - <3>.FlowLayout 流動式版面配置
 - <4>.CardLayout 多層式版面配置
 - <5>.BoxLayout 盒子式版面配置
 - <6>.GridBagLayout 進階方格式版面配置



使用 BorderLayout 版面配置→預設

1. Frame 和 JFrame 的 ContentPane 預設使用的版面編排就是 BorderLayout
2. 這種編排方式就是將一個收納器(如 Frame 視窗)內的元件排成 東·西·南·北·中 這 5 個區域
3. 這些區域永遠都會填滿此收納器，當收納器的尺寸改變時，它們仍填滿收納器，也就是此收納器內元件的尺寸大小會跟著改變，但是主要改變中區的大小

```
public class Frame0201BorderLayout1 extends JFrame {

    public JButton JB1, JB2, JB3, JB4, JB5;
    public Container ContentPane;

    public Frame0201BorderLayout1() {
        initComponents();
    }

    private void initComponents() {

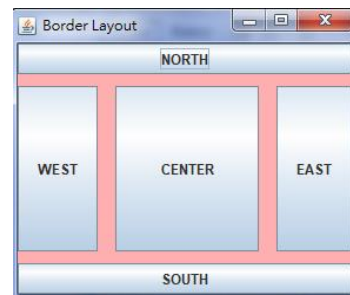
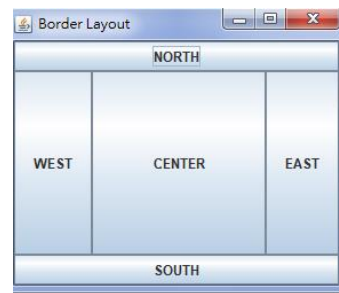
        ContentPane = this.getContentPane();
        //BorderLayout layout = new BorderLayout();
        //15=水平間距，10= 垂直間距
        BorderLayout layout = new BorderLayout(15, 10);
        ContentPane.setLayout(layout);

        JB1 = new JButton("NORTH");
        JB2 = new JButton("WEST");
        JB3 = new JButton("CENTER");
        JB4 = new JButton("EAST");
        JB5 = new JButton("SOUTH");

        ContentPane.add(JB1, BorderLayout.NORTH);
        ContentPane.add(JB2, BorderLayout.WEST);
        ContentPane.add(JB3, BorderLayout.CENTER);
        ContentPane.add(JB4, BorderLayout.EAST);
        ContentPane.add(JB5, BorderLayout.SOUTH);

        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 300, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0201BorderLayout1();
    }
}
```



BorderLayout - 屬性	
Properties	
Horizontal Gap	10
Vertical Gap	10

JButton1 [JButton] - 屬性	
Properties Binding Events Code	
verticalAlignment	CENTER
verticalTextPosition	CENTER
Layout	
Direction	Center
Accessibility	
Accessible Name	jButton1
Accessible Description	
Accessible Parent	Form

[JFrame] - 屬性	
Properties Binding Events Code	
Code Generation	
Bean Class	class javax.swing.JFrame
Form Size	[416, 338]
Form Position	[0, 0]
Form Size Policy	Generate Resize Code
Generate Position	<input checked="" type="checkbox"/>
Generate Size	<input checked="" type="checkbox"/>
Generate Center	<input type="checkbox"/>
Designer Size	[400, 300]

```
public class Frame0201BorderLayout2 extends JFrame {

    public JButton JB1, JB2, JB3, JB4, JB5;
    public JPanel JP;
    public Container ContentPane;

    public Frame0201BorderLayout2() {
        initComponents();
    }

    private void initComponents() {
        ContentPane = this.getContentPane();
        BorderLayout layout = new BorderLayout(15, 10);
        ContentPane.setLayout(null);

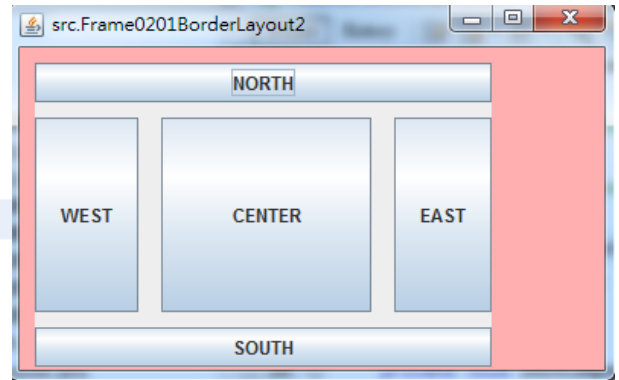
        JB1 = new JButton("NORTH");
        JB2 = new JButton("WEST");
        JB3 = new JButton("CENTER");
        JB4 = new JButton("EAST");
        JB5 = new JButton("SOUTH");

        JP = new JPanel();
        JP.setLayout(layout);
        JP.setBounds(10, 10, 300, 200);
        ContentPane.add(JP);

        JP.add(JB1, BorderLayout.NORTH);
        JP.add(JB2, BorderLayout.WEST);
        JP.add(JB3, BorderLayout.CENTER);
        JP.add(JB4, BorderLayout.EAST);
        JP.add(JB5, BorderLayout.SOUTH);

        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 400, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0201BorderLayout2();
    }
}
```



使用 GridLayout 版面配置

```
public class Frame0202GridLayout1 extends JFrame {

    public JButton JB1, JB2, JB3, JB4, JB5;
    public Container ContentPane;

    public Frame0202GridLayout1() {
        initComponents();
    }

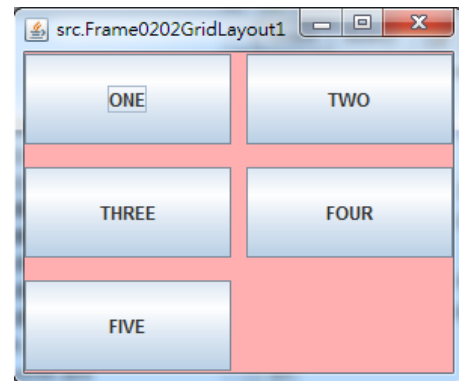
    private void initComponents() {
        ContentPane = this.getContentPane();
        //GridLayout layout = new GridLayout(2, 0 );
        //GridLayout layout = new GridLayout(0, 2);
        // GridLayout layout = new GridLayout(3, 2);
        // 3: 水平切割欄數 2 : 垂直分割列數 10: 水平間距 15 : 垂直間距
        GridLayout layout = new GridLayout(3, 2, 10, 15);
        ContentPane.setLayout(layout);

        JB1 = new JButton("ONE");
        JB2 = new JButton("TWO");
        JB3 = new JButton("THREE");
        JB4 = new JButton("FOUR");
        JB5 = new JButton("FIVE");

        ContentPane.add(JB1);
        ContentPane.add(JB2);
        ContentPane.add(JB3);
        ContentPane.add(JB4);
        ContentPane.add(JB5);

        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 300, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0202GridLayout1();
    }
}
```



```

public class Frame0202GridLayout2 extends JFrame {

    public JButton JB1, JB2, JB3, JB4, JB5;
    public Container ContentPane;
    public JPanel JP;

    public Frame0202GridLayout2() {
        initComponents();
    }

    private void initComponents() {
        ContentPane = this.getContentPane();
        GridLayout layout = new GridLayout(1, 0, 10, 15);

        JP = new JPanel();
        JB1 = new JButton("ONE");
        JB2 = new JButton("TWO");
        JB3 = new JButton("THREE");
        JB4 = new JButton("FOUR");
        JB5 = new JButton("FIVE");

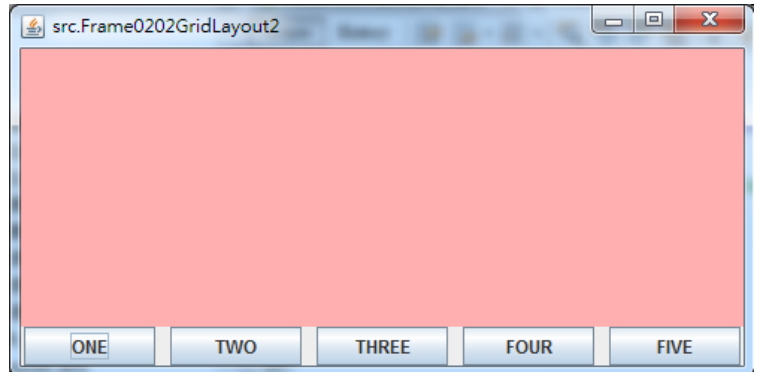
        JP.setLayout(layout);
        ContentPane.add(JP, BorderLayout.SOUTH);

        JP.add(JB1);
        JP.add(JB2);
        JP.add(JB3);
        JP.add(JB4);
        JP.add(JB5);

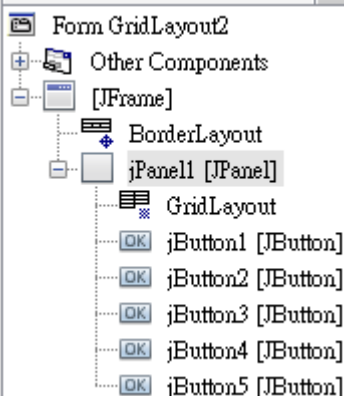
        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 500, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0202GridLayout2();
    }
}

```



jPanel1 [JPanel] - 監視器



GridLayout - 属性

Properties	
Columns	0
Horizontal Gap	15
Rows	1
Vertical Gap	10

Layout	
Direction	South

使用 FlowLayout 版面配置

由左而右、由上而下地排列元件

```

public class Frame0203FlowLayout1 extends JFrame {

    public JButton JB1, JB2, JB3, JB4, JB5;
    public Container ContentPane;

    public Frame0203FlowLayout1() {
        initComponents();
    }

    private void initComponents() {
        ContentPane = this.getContentPane();

        //1=置中 2=靠右 3=對齊左邊界 4=對齊右邊界 0=靠左排列 , 預設是 1
        FlowLayout layout = new FlowLayout(2);
        //FlowLayout layout = new FlowLayout(FlowLayout.RIGHT);
        ContentPane.setLayout(layout);

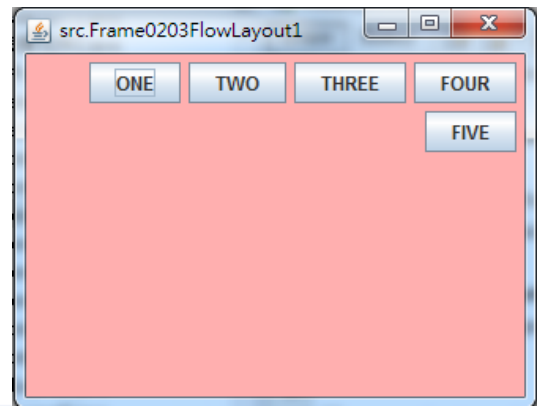
        JB1 = new JButton("ONE");
        JB2 = new JButton("TWO");
        JB3 = new JButton("THREE");
        JB4 = new JButton("FOUR");
        JB5 = new JButton("FIVE");

        ContentPane.add(JB1);
        ContentPane.add(JB2);
        ContentPane.add(JB3);
        ContentPane.add(JB4);
        ContentPane.add(JB5);

        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 300, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0203FlowLayout1();
    }
}

```



FlowLayout - 屬性	
Properties	
Align On Baseline	<input type="checkbox"/>
Alignment	Center
Horizontal Gap	5
Vertical Gap	5

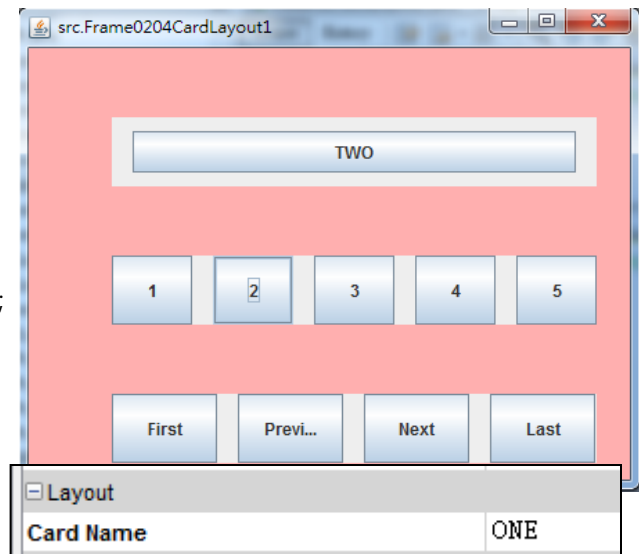
使用 CardLayout 版面配置

1. 將所有元件重疊在一起，後加入的元件放置在第一個元件之後
2. 控制顯示元件項目

ContentPane.add(元件，名稱字串)

```
public class Frame0204CardLayout1 extends JFrame {
    public JButton JB1 · JB2 · JB3 · JB4 · JB5;
    public JButton btn1 · btn2 · btn3 · btn4 · btn5;
    public JButton btnfirst · btnprevious · btnnext · btnlast;
    public JPanel JP1 · JP2 · JP3;
    public Container ContentPane;
    public CardLayout layout1 = new CardLayout(15 · 10);
    public GridLayout layout2 = new GridLayout(1 · 0 · 15 · 10);
    public Frame0204CardLayout1() {
        initComponents();
    }
}
```

```
private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(null);
    JP1 = new JPanel();
    JP2 = new JPanel();
    JP3 = new JPanel();
    JB1 = new JButton("ONE");
    JB2 = new JButton("TWO");
    JB3 = new JButton("THREE");
    JB4 = new JButton("FOUR");
    JB5 = new JButton("FIVE");
    btn1 = new JButton("1");
    btn2 = new JButton("2");
    btn3 = new JButton("3");
    btn4 = new JButton("4");
    btn5 = new JButton("5");
    btnfirst = new JButton("First");
    btnprevious = new JButton("Previous");
    btnnext = new JButton("Next");
    btnlast = new JButton("Last");
    JP1.setLayout(layout1);
    JP2.setLayout(layout2);
    JP3.setLayout(layout2);
    JP1.setBounds(60 · 50 · 350 · 50);
    JP2.setBounds(60 · 150 · 350 · 50);
    JP3.setBounds(60 · 250 · 350 · 50);
    btn1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            btn1ActionPerformed(evt);
        }
    });
    btn2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            btn2ActionPerformed(evt);
        }
    });
}
```



```
public class Frame0204CardLayout1 extends javax.swing.JFrame {

    CardLayout layout1;

    public Frame0204CardLayout1() {
        initComponents();

        layout1 = (CardLayout) JP1.getLayout();
    }
}
```

```
});  
btn3.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btn3ActionPerformed(evt);  
    }  
});  
btn4.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btn4ActionPerformed(evt);  
    }  
});  
btn5.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btn5ActionPerformed(evt);  
    }  
});  
btnfirst.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btnfirstActionPerformed(evt);  
    }  
});  
btnprevious.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btnpreviousActionPerformed(evt);  
    }  
});  
bttnext.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        bttnextActionPerformed(evt);  
    }  
});  
btnlast.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btnlastActionPerformed(evt);  
    }  
});  
ContentPane.add(JP1);  
ContentPane.add(JP2);  
ContentPane.add(JP3);  
JP1.add(JB1 · "ONE");  
JP1.add(JB2 · "TWO");  
JP1.add(JB3 · "THREE");  
JP1.add(JB4 · "FOUR");  
JP1.add(JB5 · "FIVE");  
JP2.add(btn1);  
JP2.add(btn2);  
JP2.add(btn3);  
JP2.add(btn4);  
JP2.add(btn5);  
JP3.add(btnfirst);
```

```
JP3.add(btnprevious);
JP3.add(btnnext);
JP3.add(btnlast);
ContentPane.setBackground(Color.PINK);
this.setTitle(getClass().getName().toString());
this.setBounds(200 · 100 · 450 · 350);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
private void btn1ActionPerformed(java.awt.event.ActionEvent evt) {
    layout1.show(JP1 · "ONE");
}
private void btn2ActionPerformed(java.awt.event.ActionEvent evt) {
    layout1.show(JP1 · "TWO");
}
private void btn3ActionPerformed(java.awt.event.ActionEvent evt) {
    layout1.show(JP1 · "THREE");
}
private void btn4ActionPerformed(java.awt.event.ActionEvent evt) {
    layout1.show(JP1 · "FOUR");
}
private void btn5ActionPerformed(java.awt.event.ActionEvent evt) {
    layout1.show(JP1 · "FIVE");
}
private void btnfirstActionPerformed(java.awt.event.ActionEvent evt) {
    layout1.first(JP1);
}
private void btnpreviousActionPerformed(java.awt.event.ActionEvent evt) {
    layout1.previous(JP1);
}
private void btnnextActionPerformed(java.awt.event.ActionEvent evt) {
    layout1.next(JP1);
}
private void btnlastActionPerformed(java.awt.event.ActionEvent evt) {
    layout1.last(JP1);
}
public static void main(String[] args) {
    new Frame0204CardLayout1();
}
}
```


使用 BorderLayout 版面配置

將元件放入一列或一行中，即使填滿了也不會移至下一列或下一行

X.AXIS=由左到右，Y.AXIS =由上到下

BoxLayout(Container · int)→Container 是使用的容器物件，int 是指定排列方式是水平還是垂直

```
public class Frame0205BoxLayout1 extends JFrame {

    public JButton JB1, JB2, JB3, JB4, JB5;
    public Container ContentPane;

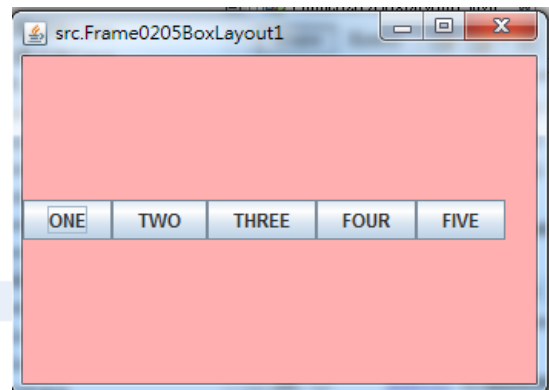
    public Frame0205BoxLayout1() {
        initComponents();
    }

    private void initComponents() {
        ContentPane = this.getContentPane();
        // X.AXIS=由左到右，Y.AXIS =由上到下
        BoxLayout layout = new BoxLayout(ContentPane, BoxLayout.X_AXIS);
        ContentPane.setLayout(layout);

        JB1 = new JButton("ONE");
        JB2 = new JButton("TWO");
        JB3 = new JButton("THREE");
        JB4 = new JButton("FOUR");
        JB5 = new JButton("FIVE");
        ContentPane.add(JB1);
        ContentPane.add(JB2);
        ContentPane.add(JB3);
        ContentPane.add(JB4);
        ContentPane.add(JB5);

        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 350, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0205BoxLayout1();
    }
}
```

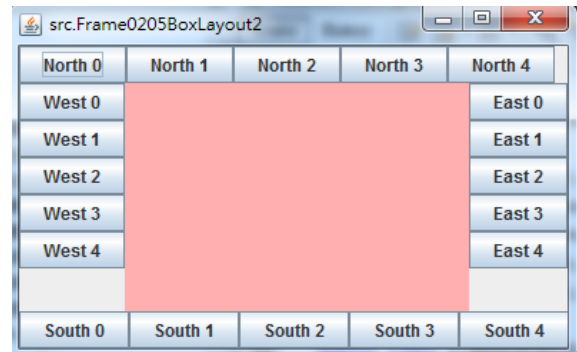


BoxLayout - 屬性 ⓘ	
Properties	
Axis	X Axis

```

public class Frame0205BoxLayout2 extends JFrame {
    public JPanel JP1, JP2, JP3, JP4;
    public Container ContentPane;
    public Frame0205BoxLayout2() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        BorderLayout layout = new BorderLayout();
        ContentPane.setLayout(layout);
        JP1 = new JPanel();
        JP2 = new JPanel();
        JP3 = new JPanel();
        JP4 = new JPanel();
        for (int i = 0; i < 5; i++) {
            JP1.add(new JButton("West " + i));
        }
        for (int i = 0; i < 5; i++) {
            JP2.add(new JButton("East " + i));
        }
        for (int i = 0; i < 5; i++) {
            JP3.add(new JButton("North " + i));
        }
        for (int i = 0; i < 5; i++) {
            JP4.add(new JButton("South " + i));
        }
        JP1.setLayout(new BoxLayout(JP1, BoxLayout.Y_AXIS));
        JP2.setLayout(new BoxLayout(JP2, BoxLayout.Y_AXIS));
        JP3.setLayout(new BoxLayout(JP3, BoxLayout.X_AXIS));
        JP4.setLayout(new BoxLayout(JP4, BoxLayout.X_AXIS));
        ContentPane.add(JP1, BorderLayout.WEST);
        ContentPane.add(JP2, BorderLayout.EAST);
        ContentPane.add(JP3, BorderLayout.NORTH);
        ContentPane.add(JP4, BorderLayout.SOUTH);
        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 400, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
    public static void main(String[] args) {
        new Frame0205BoxLayout2();
    }
}

```



使用 GridBagLayout 版面配置

GridBagLayout 格子袋式版面配置屬於比較複雜的版面配置，類似 GridLayout 可以將元件分為欄和列排列，不過列和欄並不需等高或等寬，當視窗設定為 GridBagLayout 的排版方式後，視窗物件利用 add() 方法加入視窗元件時，必須額外提供 GridBagConstraints 類別 (class) 的參數 (parameter)，因為實際的設定是由 GridBagConstraints 型態的物件 (object) 來完成的

成員變數	說明
gridx gridy	指定元件放置的位置是在哪一欄和哪一行，左上角的欄 gridx 為 0，列 gridy 為 0
gridwidth gridheight	指定元件放置的位置佔用幾欄和幾行，預設值為 1，相關常數的說明，如下所示： GridBagConstraints.REMSINDER：欄或行的最後 1 個。 GridBagConstraints.RELATIVE：欄或行中下一個到最後。
fill	調整元件在顯示區域的尺寸，相關常數的說明，如下所示： GridBagConstraints.NONE：不需要，預設值。 GridBagConstraints.HORIZONTAL：調整寬度填滿顯示區域，但不更改高度。 GridBagConstraints.VERTICAL：調整高度填滿顯示區域，但不更改寬度。 GridBagConstraints.BOTH：調整元件填滿顯示區域。
ipadx ipady	指定內部填充的距離，預設值為 0，填充是套用在元件的 2 邊
insets	指定元件的外部填充，預設沒有，這是元件和顯示區域邊界的距離
anchor (定位點)	元件如果小於顯示區域，可以指定元件位置，相關常數的說明，如下所示：(如果 fill 是 both 則無效用) GridBagConstraints.CENTER：置中，預設值。 GridBagConstraints.NORTH：北。 GridBagConstraints.NORTHEAST：東北。 GridBagConstraints.EAST：東。 GridBagConstraints.SOUTHEAST：東南。 GridBagConstraints.SOUTH：南。 GridBagConstraints.SOUTHWEST：西南。 GridBagConstraints.WEST：西。 GridBagConstraints.NORTHWEST：西北。
weightx weighty	設定欄和行中元件分配空間的權值，其範圍是 0.0~1.0，值愈大表示元件在欄和行中可以取得更多空間

```

public class Frame0206GridBagLayout1 extends JFrame {

    public JButton[] JB = new JButton[5];
    public Container ContentPane;

    public Frame0206GridBagLayout1() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        GridBagLayout layout = new GridBagLayout();
        ContentPane.setLayout(layout);
        GridBagConstraints g1 = new GridBagConstraints();
        g1.fill = GridBagConstraints.BOTH; // 當格子有剩餘空間時，填充空間
        g1.gridx = 0; // 元件在方格中的水平座標
        g1.gridy = 0; // 元件在方格中的垂直座標
        g1.gridwidth = 3; // 元件所佔水平區域大小
        g1.gridheight = 1; // 元件所佔垂直區域大小
        g1.weightx = 1.0; // 當視窗放大時，長度不變
        g1.weighty = 1.0; // 當視窗放大時，高度不變
        g1.insets = new Insets(6, 6, 6, 6); // 組件彼此的間距
        JB[0] = new JButton("按鈕1");
        ContentPane.add(JB[0], g1);
        for (int i = 0; i <= 3; i++) {
            JB[i] = new JButton("按鈕 " + String.valueOf(i));
            g1.gridx = i; // 元件在方格中的水平座標
            g1.gridy = 1; // 元件在方格中的垂直座標
            g1.gridwidth = 1; // 元件所佔水平區域大小
            g1.gridheight = 1; // 元件所佔垂直區域大小
            ContentPane.add(JB[i], g1);
        }
        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 400, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0206GridBagLayout1();
    }
}

```



Layout	
Grid X	0
Grid Y	0
Grid Width	3
Grid Height	1
Fill	Both
Internal Padding X	0
Internal Padding Y	0
Anchor	West
Weight X	1.0
Weight Y	1.0
Insets	[6, 6, 6, 6]

```

public class Frame0206GridBagLayout2 extends JFrame {

    public Container ContentPane;

    public JLabel JL1 · JL2 · JL3;

    public JTextField JT1 · JT2;

    public JButton JB1 · JB2 · JB3 · JB4 · JB5 · JB6 · JB7;

    public JButton[] JB;

    public Frame0206GridBagLayout2() {
        initComponents();
    }

    private void initComponents() {

        ContentPane = this.getContentPane();
        ContentPane.setLayout(new GridBagLayout());

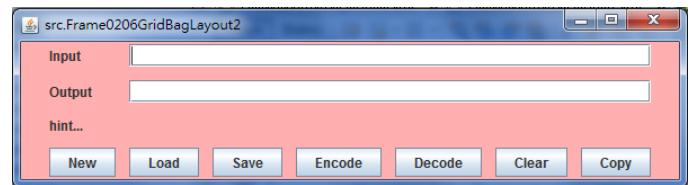
        JL1 = new JLabel("Input");
        JL2 = new JLabel("Output");
        JL3 = new JLabel("hint...");
        JT1 = new JTextField();
        JT2 = new JTextField();
        JB1 = new JButton("New");
        JB2 = new JButton("Load");
        JB3 = new JButton("Save");
        JB4 = new JButton("Encode");
        JB5 = new JButton("Decode");
        JB6 = new JButton("Clear");
        JB7 = new JButton("Copy");

        //Input
        GridBagConstraints c0 = new GridBagConstraints();
        c0.gridx = 0;
        c0.gridy = 0;
        c0.gridwidth = 1;
        c0.gridheight = 1;
        c0.insets = new Insets(6 · 6 · 6 · 6);
        ContentPane.add(JL1 · c0);

        //Output
        GridBagConstraints c1 = new GridBagConstraints();
        c1.gridx = 0;
        c1.gridy = 1;
        c1.gridwidth = 1;
        c1.gridheight = 1;
        c1.insets = new Insets(6 · 6 · 6 · 6);
        ContentPane.add(JL2 · c1);

        //hint...
        GridBagConstraints c2 = new GridBagConstraints();
        c2.gridx = 0;

```



Layout	
Grid X	1
Grid Y	0
Grid Width	6
Grid Height	1
Fill	Horizontal
Internal Padding X	0
Internal Padding Y	0
Anchor	Center
Weight X	0.0
Weight Y	0.0
Insets	[6, 6, 6, 6]

```
c2.gridy = 2;
c2.gridwidth = 1;
c2.gridheight = 1;
c2.insets = new Insets(6 · 6 · 6 · 6);
ContentPane.add(JL3 · c2);
//Input 文字方塊
GridBagConstraints c3 = new GridBagConstraints();
c3.gridx = 1;
c3.gridy = 0;
c3.gridwidth = 6;
c3.gridheight = 1;
c3.fill = GridBagConstraints. HORIZONTAL;
c3.insets = new Insets(6 · 6 · 6 · 6);
ContentPane.add(JT1 · c3);
//Output 文字方塊
GridBagConstraints c4 = new GridBagConstraints();
c4.gridx = 1;
c4.gridy = 1;
c4.gridwidth = 6;
c4.gridheight = 1;
c4.fill = GridBagConstraints. HORIZONTAL;
c4.insets = new Insets(6 · 6 · 6 · 6);
ContentPane.add(JT2 · c4);
//按鈕
JB = new JButton[]{JB1 · JB2 · JB3 · JB4 · JB5 · JB6 · JB7};
for (int i = 0; i < 7; i++) {
    GridBagConstraints c5 = new GridBagConstraints();
    c5.gridx = i;
    c5.gridy = 3;
    c5.gridwidth = 1;
    c5.gridheight = 1;
    c5.insets = new Insets(6 · 6 · 6 · 6);
    ContentPane.add(JB[i] · c5);
}
ContentPane.setBackground(Color.PINK);
this.setTitle(getClass().getName().toString());
this.setBounds(200 · 100 · 600 · 160);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
public static void main(String[] args) {
    new Frame0206GridBagLayout2();
}
```

}

```
public class Frame0207預設Layout extends JFrame {

    public JButton JB1;
    public Container ContentPane;

    public Frame0207預設Layout() {
        initComponents();
    }

    private void initComponents() {
        ContentPane = this.getContentPane();
        //預設使用的是 BorderLayout
        // this.setLayout(null);
        JB1 = new JButton("OK");
        JB1.setBounds(20, 80, 150, 30);
        ContentPane.add(JB1); //沒指定位置， 預設是 CENTER

        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200, 100, 300, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        new Frame0207預設Layout();
    }
}
```



//Window 類別 (class) 的 pack() 方法 (method) 用來將視窗中的各個元件，自動依版面配置作適當的排列

```
public class Frame0208pack方法 extends JFrame {
```

```
    public JButton JB1;  
    public JButton JB2;  
    public JButton JB3;  
    public JButton JB4;  
    public JButton JB5;
```

```
    public Container ContentPane;
```

```
    public Frame0208pack方法() {  
        initComponents();  
    }
```

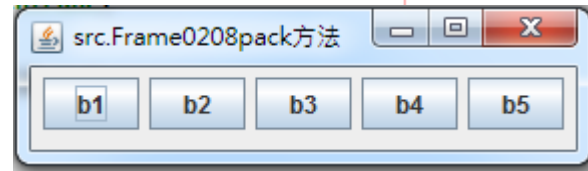
```
    private void initComponents() {  
        ContentPane = this.getContentPane();  
        ContentPane.setLayout(new FlowLayout());
```

```
        JB1 = new JButton("b1");  
        JB2 = new JButton("b2");  
        JB3 = new JButton("b3");  
        JB4 = new JButton("b4");  
        JB5 = new JButton("b5");  
        ContentPane.add(JB1);  
        ContentPane.add(JB2);  
        ContentPane.add(JB3);  
        ContentPane.add(JB4);  
        ContentPane.add(JB5);
```

```
        this.setTitle(getClass().getName().toString());  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.pack();  
        // this.setLocation(300, 300);  
        // this.setSize(400, 400);  
        this.setVisible(true);  
    }
```

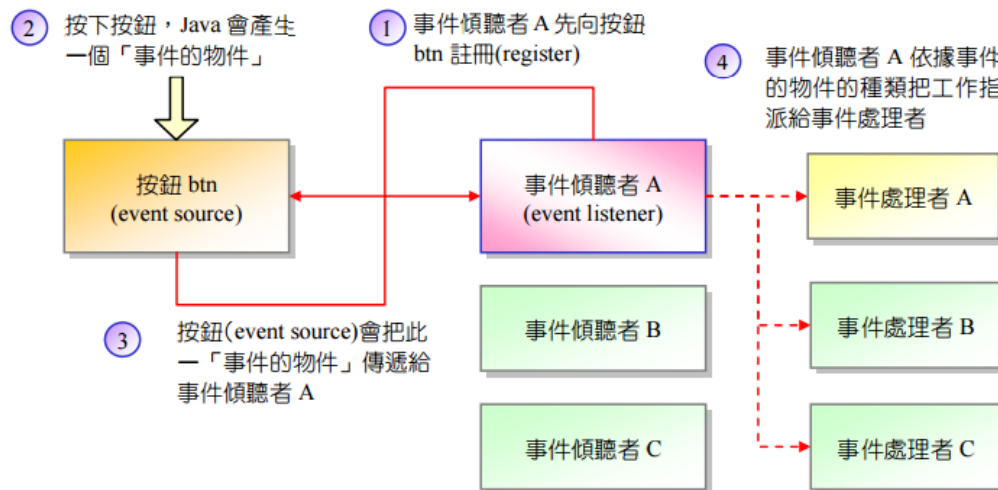
```
    public static void main(String[] args) {  
        new Frame0208pack方法();  
    }
```

```
}
```



事件的基本觀念

1. 圖形使用者介面(GUI)的程式是採用事件驅動(Event Drive)，例如一個 GUI 的應用程式畫面上有一個 Button，當使用者利用滑鼠游標在 Button 上面按一下時，Button 物件就會產生事件，而程式會對事件進行反應，以達到使用者與程式間的互動關係
2. Java 在 1.1 版之後對於事件的處理方式採用“委派事件模型”，當事件發生時，產生事件的物件會把此一「訊息」轉給「事件傾聽者」(event listener)處理的方式，它的特色是 GUI 元件本身通常不直接負責處理起源自它的事件，而是「委任」給特定的 Listener 物件(事件傾聽者)來處理，
3. 首先得讓視窗元件傾聽所有使用者可能的動作，依據使用者的動作，例如在文字方塊(textfield)中輸入文字，我們設計程式相對應的動作，如可以在另一個標籤(label)中顯示提示訊息等因此，我們得要在視窗中實作(implement)傾聽者介面，然後實作出相對應的方法，最後視窗元件得登記它要傾聽事件(addxxxListener)。



4. 建立 GUI 事件的作法，主要有

<1>.實作 Listener 介面的作法----implements XXXListener

<2>.繼承 Adapter 類別的作法----extends XXXAdapter

Standard Ed. 6
[全部類別](#)
 軟體套件
[java.applet](#)
[java.awt](#)
[java.awt.color](#)
[java.awt.datatransfer](#)
[java.awt.dnd](#)
[java.awt.event](#)
[java.awt.font](#)
[java.awt.geom](#)
[java.awt.im](#)
[java.awt.image](#)
 介面
[ActionListener](#)
[AdjustmentListener](#)
[AWTEventListener](#)
[ComponentListener](#)
[ContainerListener](#)
[FocusListener](#)
[HierarchyBoundsListener](#)
[HierarchyListener](#)
[InputMethodListener](#)
[ItemListener](#)
[KeyListener](#)
[MouseListener](#)
[MouseMotionListener](#)
[MouseWheelListener](#)
[TextListener](#)
[WindowFocusListener](#)
[WindowListener](#)
[WindowStateListener](#)

1.1
 另請參見：
[MouseAdapter](#), [MouseEvent](#), [Tutorial: Writing a Mouse Listener](#)
方法摘要

void	mouseClicked(MouseEvent e)	鼠標按鍵在元件上單擊〈按下並釋放〉時調用。
void	mouseEntered(MouseEvent e)	鼠標進入到元件上時調用。
void	mouseExited(MouseEvent e)	鼠標離開元件時調用。
void	mousePressed(MouseEvent e)	鼠標按鍵在元件上按下時調用。
void	mouseReleased(MouseEvent e)	鼠標按鈕在元件上釋放時調用。

方法詳細資訊
mouseClicked
 void mouseClicked([MouseEvent e](#))
 鼠標按鍵在元件上單擊〈按下並釋放〉時調用。

軟體套件
[java.applet](#)
[java.awt](#)
[java.awt.color](#)
[java.awt.datatransfer](#)
[java.awt.dnd](#)
[java.awt.event](#)
[java.awt.font](#)
[java.awt.geom](#)
[java.awt.im](#)
[java.awt.im.spi](#)
[java.awt.image](#)
[java.awt.image.renderable](#)

[InputMethodListener](#)
[ItemListener](#)
[KeyListener](#)
[MouseListener](#)
[MouseMotionListener](#)
[MouseWheelListener](#)
[TextListener](#)
[WindowFocusListener](#)
[WindowListener](#)
[WindowStateListener](#)

類別
[ActionEvent](#)
[AdjustmentEvent](#)
[AWTEventListenerProxy](#)
[ComponentAdapter](#)
[ComponentEvent](#)
[ContainerAdapter](#)
[ContainerEvent](#)
[FocusAdapter](#)
[FocusEvent](#)
[HierarchyBoundsAdapter](#)
[HierarchyEvent](#)
[InputEvent](#)
[InputMethodEvent](#)
[InvocationEvent](#)
[ItemEvent](#)
[KeyAdapter](#)
[KeyEvent](#)
[MouseAdapter](#)
[MouseEvent](#)
[MouseMotionAdapter](#)
[MouseWheelEvent](#)

建構子摘要

[MouseAdapter](#) ()

方法摘要

void	mouseClicked (MouseEvent e)	鼠標按鍵在元件上單擊（按下並釋放）時調用。
void	mouseDragged (MouseEvent e)	鼠標按鍵在元件上按下並拖動時調用。
void	mouseEntered (MouseEvent e)	鼠標進入到元件上時調用。
void	mouseExited (MouseEvent e)	鼠標離開元件時調用。
void	mouseMoved (MouseEvent e)	鼠標鼠標移動到元件上但無按鍵按下時調用。
void	mousePressed (MouseEvent e)	鼠標按鍵在元件上按下時調用。
void	mouseReleased (MouseEvent e)	鼠標按鈕在元件上釋放時調用。
void	mouseWheelMoved (MouseWheelEvent e)	鼠標滾輪旋轉時調用。

從類別 [java.lang.Object](#) 繼承的方法

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#),

建構子詳細資訊

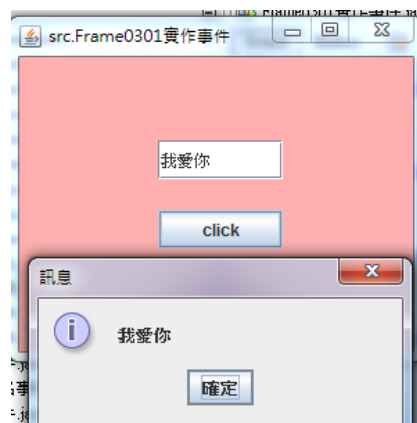
MouseAdapter

public [MouseAdapter](#)()

實作 Listener 介面的步驟

1. 自訂一個實作 XXXListener 的類別 → myXXXListener
2. 加入所需的傾聽能力 → 元件.addXXXListener
3. 產生 XXXListener 物件實體

```
public class Frame0301 實作事件 extends JFrame {
    public JTextField JT1;
    public JButton JB1;
    public Container ContentPane;
    public Frame0301 實作事件() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        JT1 = new JTextField("我愛你");
        JB1 = new JButton("click");
        JT1.setBounds(100, 60, 90, 28);
        JB1.setBounds(100, 110, 90, 28);
        JB1.addMouseListener(new myMouseListener());
        ContentPane.add(JT1);
        ContentPane.add(JB1);
    }
}
```



```
ContentPane.setBackground(Color.PINK);
this.setTitle(getClass().getName().toString());
this.setBounds(200 · 100 · 300 · 250);
this.addWindowListener(new myWindowListener());
// this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

public class myMouseListener implements MouseListener {
    public void mouseClicked(MouseEvent e) {
        JOptionPane.showMessageDialog(JB1 · JT1.getText().trim());
    }
    public void mouseEntered(MouseEvent e) {
    }
    public void mouseExited(MouseEvent e) {
    }
    public void mousePressed(MouseEvent e) {
    }
    public void mouseReleased(MouseEvent e) {
    }
}

public class myWindowListener implements WindowListener {
    public void windowActivated(WindowEvent e) {
    }
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
    public void windowOpened(WindowEvent e) {
    }
    public void windowDeiconified(WindowEvent e) {
    }
    public void windowDeactivated(WindowEvent e) {
    }
    public void windowIconified(WindowEvent e) {
    }
    public void windowClosed(WindowEvent e) {
    }
}

public static void main(String[] args) {
    new Frame0301 實作事件();
}
}
```

繼承 Adapter 類別的步驟

1. 若 XXXListener 介面中的事件函數有 兩個以上(含兩個)，Java 就有提供一個對應它的 XXXAdapter 類別(即改編者類別)給我們
2. 它就是一個已經 implements XXXListener 的類別，不過 XXXAdapter 類別之事件函數的實作部份都是空的
3. 加入所需的傾聽能力→元件.addXXXListener
4. 自訂一個繼承 XXXAdapter 的類別---extends XXXAdapter
5. 產生 XXXAdapter 物件實體

此種作法只須覆寫要用到的事件函數 mouseClicked(MouseEvent e)→此種作法需要寫的程式碼較少

```
public class Frame0302 繼承事件 extends JFrame {  
    public JTextField JT1;  
    public JButton JB1;  
    public Container ContentPane;  
    public Frame0302 繼承事件() {  
        initComponents();  
    }  
    private void initComponents() {  
        ContentPane = this.getContentPane();  
        ContentPane.setLayout(null);  
        JT1 = new JTextField("我愛你");  
        JB1 = new JButton("click");  
        JT1.setBounds(100, 60, 90, 28);  
        JB1.setBounds(100, 110, 90, 28);  
        JB1.addMouseListener(new myMouseAdapter());  
        ContentPane.add(JT1);  
        ContentPane.add(JB1);  
        ContentPane.setBackground(Color.PINK);  
        this.setTitle(getClass().getName());  
        this.setBounds(200, 100, 300, 250);  
        this.addWindowListener(new myWindowAdapter());  
        // this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setVisible(true);  
    }  
    public class myMouseAdapter extends MouseAdapter {  
        public void mouseClicked(MouseEvent e) {  
            JOptionPane.showMessageDialog(JB1, JT1.getText().trim());  
        }  
    }  
    public class myWindowAdapter extends WindowAdapter {  
        public void windowClosing(WindowEvent e) {  
            System.exit(0);  
        }  
    }  
    public static void main(String[] args) {  
        new Frame0302 繼承事件();  
    }  
}
```

運用匿名類別

1. 我們自訂的 Lister 類別若只需制定一、兩個事件函數，就很適合使用匿名類別

```
public class Frame0303 匿名事件 extends JFrame {
    public JTextField JT1;
    public JButton JB1;
    public Container ContentPane;
    public Frame0303 匿名事件() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        JT1 = new JTextField("我愛你");
        JB1 = new JButton("click");
        JT1.setBounds(100, 60, 90, 28);
        JB1.setBounds(100, 110, 90, 28);
        JB1.addMouseListener(new MouseAdapter() {
            public void mouseClicked(MouseEvent e) {
                JOptionPane.showMessageDialog(JB1, JT1.getText().trim());
            }
        });
        ContentPane.add(JT1);
        ContentPane.add(JB1);
        //////////////////////////////////////
        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName());
        this.setBounds(200, 100, 300, 250);
        this.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        // this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
    public static void main(String[] args) {
        new Frame0303 匿名事件();
    }
}
```

```
public class Frame0304 預設事件 extends JFrame {
    public JTextField JT1;
    public JButton JB1;
    public Container ContentPane;
    public Frame0304 預設事件() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        JT1 = new JTextField("我愛你");
        JB1 = new JButton("click");
        JT1.setBounds(100 · 60 · 90 · 28);
        JB1.setBounds(100 · 110 · 90 · 28);
        JB1.addActionListener(new myActionListener());
        ContentPane.add(JT1);
        ContentPane.add(JB1);
        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName());
        this.setBounds(200 · 100 · 300 · 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
    class myActionListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(JB1 · JT1.getText().trim());
        }
    }
    public static void main(String[] args) {
        new Frame0304 預設事件();
    }
}
```

```
public class Frame0305 預設匿名事件 extends JFrame {
    public JTextField JT1;
    public JButton JB1;
    public Container ContentPane;
    public Frame0305 預設匿名事件() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        JT1 = new JTextField("我愛你");
        JB1 = new JButton("click");
        JT1.setBounds(100, 60, 90, 28);
        JB1.setBounds(100, 110, 90, 28);
        JB1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(JB1, JT1.getText().trim());
            }
        });
        ContentPane.add(JT1);
        ContentPane.add(JB1);
        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName());
        this.setBounds(200, 100, 300, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
    public static void main(String[] args) {
        new Frame0305 預設匿名事件();
    }
}
```

```
public class Frame0306 滑鼠事件 extends JFrame {
```



```
public JTextField JT1;
public JButton JB1 · JB2 · JB3;
public Container ContentPane;
public JPanel JP1;
public Frame0306 滑鼠事件() {
    initComponents();
}
private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(null);
    GridLayout layout = new GridLayout(1 · 0);
    layout.setHgap(15);
    JT1 = new JTextField();
    JP1 = new JPanel();
    JB1 = new JButton("我愛你");
    JB2 = new JButton("清除");
    JB3 = new JButton("結束");
    JT1.setBounds(170 · 30 · 100 · 28);
    JP1.setBounds(20 · 100 · 400 · 28);
    JP1.setLayout(layout);
    JP1.add(JB1);
    JP1.add(JB2);
    JP1.add(JB3);
    JB1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            JT1.setText("我愛你");
        }
    });
    JB2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            JT1.setText("");
        }
    });
    JB3.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            System.exit(0);
        }
    });
    ContentPane.add(JT1);
    ContentPane.add(JP1);
    //=====
    ContentPane.setBackground(Color.PINK);
    this.setBounds(200 · 100 · 500 · 200);
}
```



```
this.setTitle(getClass().getName());  
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
this.setVisible(true);  
}  
public static void main(String[] args) {  
    new Frame0306 滑鼠事件();  
}  
}
```

```
public class Frame0307 滑鼠個別副程式事件 extends JFrame {  
    public JTextField JT1;  
    public JButton JB1 · JB2 · JB3;  
    public Container ContentPane;  
    public JPanel JP1;  
    public Frame0307 滑鼠個別副程式事件() {  
        initComponents();  
    }  
    private void initComponents() {  
        ContentPane = this.getContentPane();  
        ContentPane.setLayout(null);  
        GridLayout layout = new GridLayout(1 · 0);  
        layout.setHgap(15);  
        JT1 = new JTextField();  
        JP1 = new JPanel();  
        JB1 = new JButton("我愛你");  
        JB2 = new JButton("清除");  
        JB3 = new JButton("結束");  
        JT1.setBounds(170 · 30 · 100 · 28);  
        JP1.setBounds(20 · 100 · 400 · 28);  
        JB1.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent evt) {  
                JB1actionPerformed(evt);  
            }  
        });  
        JB2.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent evt) {  
                JB2actionPerformed(evt);  
            }  
        });  
        JB3.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent evt) {  
                JB3actionPerformed(evt);  
            }  
        });  
        ContentPane.add(JT1);  
        JP1.setLayout(layout);  
        ContentPane.add(JP1);  
        JP1.add(JB1);  
        JP1.add(JB2);  
        JP1.add(JB3);  
    }  
}
```

```
//=====
ContentPane.setBackground(Color.PINK);
this.setBounds(200 · 100 · 500 · 200);
this.setTitle(getClass().getName());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
public void JB1ActionPerformed(ActionEvent evt) {
    JT1.setText("我愛你");
}
public void JB2ActionPerformed(ActionEvent evt) {
    JT1.setText("");
}
public void JB3ActionPerformed(ActionEvent evt) {
    System.exit(0);
}
public static void main(String[] args) {
    new Frame0307 滑鼠個別副程式事件();
}
}
```

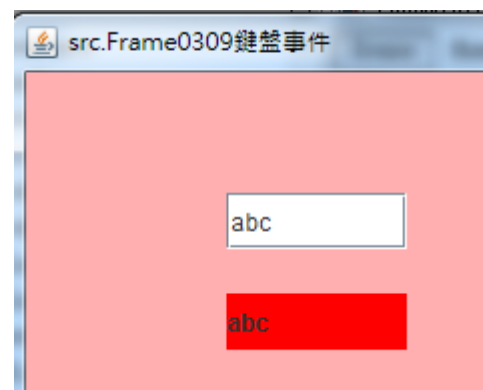
```
public class Frame0308 滑鼠共用副程式事件 extends JFrame {
    public JTextField JT1;
    public JButton JB1 · JB2 · JB3;
    public Container ContentPane;
    public JPanel JP1;
    public Frame0308 滑鼠共用副程式事件() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        GridLayout layout = new GridLayout(1 · 0);
        layout.setHgap(15);
        JT1 = new JTextField();
        JP1 = new JPanel();
        JB1 = new JButton("我愛你");
        JB2 = new JButton("清除");
        JB3 = new JButton("結束");
        JT1.setBounds(170 · 30 · 100 · 28);
        JP1.setBounds(20 · 100 · 400 · 28);
        JB1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                choose(evt);
            }
        });
        JB2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                choose(evt);
            }
        });
    }
}
```

```
JB3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
ContentPane.add(JT1);
JP1.setLayout(layout);
ContentPane.add(JP1);
JP1.add(JB1);
JP1.add(JB2);
JP1.add(JB3);
ContentPane.setBackground(Color.PINK);
this.setBounds(200 · 100 · 500 · 200);
this.setTitle(getClass().getName().toString());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

public void choose(ActionEvent evt) {
    JButton tmp = (JButton) evt.getSource();
    if (tmp == JB1) {
        JT1.setText("我愛你");
    }
    if (tmp == JB2) {
        JT1.setText("");
    }
    if (tmp == JB3) {
        System.exit(0);
    }
}

public static void main(String[] args) {
    new Frame0308 滑鼠共用副程式事件();
}
}
```

```
public class Frame0309 鍵盤事件 extends JFrame {
    public JTextField JT1;
    public JLabel JL1;
    public Container ContentPane;
    public Frame0309 鍵盤事件() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        JT1 = new JTextField("");
        JL1 = new JLabel("");
        JT1.setBounds(100 · 60 · 90 · 28);
        JL1.setBounds(100 · 110 · 90 · 28);
        JL1.setBackground(Color.red);
    }
}
```



```
JL1.setOpaque(true);
JT1.addKeyListener(new KeyAdapter() {
    public void keyPressed(KeyEvent evt) {
        choose(evt);
    }
});
ContentPane.add(JT1);
ContentPane.add(JL1);
ContentPane.setBackground(Color.PINK);
this.setTitle(getClass().getName());
this.setBounds(200 · 100 · 400 · 200);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

public void choose(KeyEvent evt) {
    if (evt.getKeyChar() == KeyEvent.VK_ENTER) {
        JL1.setText(JT1.getText());
    }
}

public static void main(String[] args) {
    new Frame0309 鍵盤事件();
}
}
```

```
public class Frame0310 鍵盤預設事件 extends JFrame {
    public JTextField JT1;
    public JLabel JL1;
    public Container ContentPane;
    public Frame0310 鍵盤預設事件() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        JT1 = new JTextField("");
        JL1 = new JLabel("");
        JT1.setBounds(100 · 60 · 90 · 28);
        JL1.setBounds(100 · 110 · 90 · 28);
        JL1.setBackground(Color.red);
        JL1.setOpaque(true);
        JT1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                choose(evt);
            }
        });
        ContentPane.add(JT1);
        ContentPane.add(JL1);
    }
}
```

```

        ContentPane.setBackground(Color.PINK);
        this.setTitle(getClass().getName().toString());
        this.setBounds(200 · 100 · 400 · 200);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
    public void choose(ActionEvent evt) {
        JL1.setText(JT1.getText());
    }
    public static void main(String[] args) {
        new Frame0310 鍵盤預設事件();
    }
}

```

```

public class Frame0401 控制項陣列 1 extends JFrame {
    public int[] nums; //用來存已取得的號碼
    public JTextField JT1 · JT2 · JT3 · JT4 · JT5 · JT6;
    public JButton JB1 · JB2;
    public Container ContentPane;
    public JPanel JP1;
    public Frame0401 控制項陣列 1() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        GridLayout layout = new GridLayout(1 · 0);
        layout.setHgap(15);
        JP1 = new JPanel();
        JT1 = new JTextField();
        JT2 = new JTextField();
        JT3 = new JTextField();
        JT4 = new JTextField();
        JT5 = new JTextField();
        JT6 = new JTextField();
        JB1 = new JButton("clear");
        JB2 = new JButton("select");
        JP1.setBounds(10 · 35 · 250 · 28);
        JB1.setBounds(40 · 80 · 100 · 28);
        JB2.setBounds(150 · 80 · 100 · 28);
        JB1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                JB1ActionPerformed(evt);
            }
        });
        JB2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                JB2ActionPerformed(evt);
            }
        });
        JP1.setLayout(layout);
    }
}

```



```
JP1.add(JT1);
JP1.add(JT2);
JP1.add(JT3);
JP1.add(JT4);
JP1.add(JT5);
JP1.add(JT6);
ContentPane.add(JP1);
ContentPane.add(JB1);
ContentPane.add(JB2);
JT1.setEditable(false);
JT2.setEditable(false);
JT3.setEditable(false);
JT4.setEditable(false);
JT5.setEditable(false);
JT6.setEditable(false);
ContentPane.setBackground(Color.PINK);
this.setBounds(200 · 100 · 300 · 200);
this.setTitle(getClass().getName().toString());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
private void JB1ActionPerformed(java.awt.event.ActionEvent evt) {
    JT1.setText("");
    JT2.setText("");
    JT3.setText("");
    JT4.setText("");
    JT5.setText("");
    JT6.setText("");
}
private void JB2ActionPerformed(java.awt.event.ActionEvent evt) {
    nums = new int[6];
    for (int i = 0; i < nums.length; i++) {
        nums[i] = (int) (Math.random() * 38) + 1;
        System.out.print(nums[i] + " ");
        for (int j = 0; j < i; j++) {
            if (nums[i] == nums[j]) {
                i = i - 1;
            }
        }
    }
    System.out.println();
    JT1.setText(String.valueOf(nums[0]));
    JT2.setText(String.valueOf(nums[1]));
    JT3.setText(String.valueOf(nums[2]));
    JT4.setText(String.valueOf(nums[3]));
    JT5.setText(String.valueOf(nums[4]));
    JT6.setText(String.valueOf(nums[5]));
}
public static void main(String[] args) {
    new Frame0401 控制項陣列 1();
}
```

```
}  
}
```

```
public class Frame0401 控制項陣列 2 extends JFrame {  
    public int[] nums;  
    public JPanel JP1;  
    public JTextField JT1 · JT2 · JT3 · JT4 · JT5 · JT6;  
    public JButton JB1 · JB2;  
    public Container ContentPane;  
    public JTextField[] JT;  
    public Frame0401 控制項陣列 2() {  
        initComponents();  
        JT = new JTextField[]{JT1 · JT2 · JT3 · JT4 · JT5 · JT6};  
        for (int i = 0; i < JT.length; i++) {  
            JT[i].setEditable(false);  
        }  
    }  
    private void initComponents() {  
        ContentPane = this.getContentPane();  
        ContentPane.setLayout(null);  
        GridLayout layout = new GridLayout(1 · 0);  
        layout.setHgap(15);  
        JP1 = new JPanel();  
        JT1 = new JTextField();  
        JT2 = new JTextField();  
        JT3 = new JTextField();  
        JT4 = new JTextField();  
        JT5 = new JTextField();  
        JT6 = new JTextField();  
        JB1 = new JButton("clear");  
        JB2 = new JButton("select");  
        JP1.setBounds(10 · 35 · 250 · 28);  
        JB1.setBounds(40 · 80 · 100 · 28);  
        JB2.setBounds(150 · 80 · 100 · 28);  
        JB1.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent evt) {  
                JB1ActionPerformed(evt);  
            }  
        });  
        JB2.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent evt) {  
                JB2ActionPerformed(evt);  
            }  
        });  
        JP1.setLayout(layout);  
        JP1.add(JT1);  
        JP1.add(JT2);  
        JP1.add(JT3);  
        JP1.add(JT4);  
        JP1.add(JT5);  
        JP1.add(JT6);
```

```

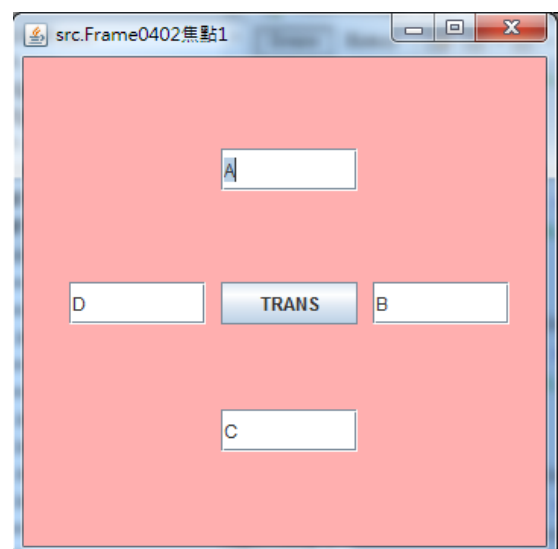
ContentPane.add(JP1);
ContentPane.add(JB1);
ContentPane.add(JB2);
ContentPane.setBackground(Color.PINK);
this.setBounds(200 · 100 · 350 · 300);
this.setTitle(getClass().getName().toString());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
private void JB1ActionPerformed(java.awt.event.ActionEvent evt) {
    for (int i = 0; i < JT.length; i++) {
        JT[i].setText("");
    }
}
private void JB2ActionPerformed(java.awt.event.ActionEvent evt) {
    nums = new int[6];
    for (int i = 0; i < nums.length; i++) {
        nums[i] = (int) (Math.random() * 38) + 1;
        System.out.print(nums[i] + " ");
        for (int j = 0; j < i; j++) {
            if (nums[i] == nums[j]) {
                i = i - 1;
            }
        }
    }
    System.out.println();
    for (int i = 0; i < JT.length; i++) {
        JT[i].setText(String.valueOf(nums[i]));
    }
}
public static void main(String[] args) {
    new Frame0401 控制項陣列 2();
}
}

```

```

public class Frame0402 焦點 1 extends JFrame {
    public JTextField JT1 · JT2 · JT3 · JT4;
    public JButton JB1;
    public Container ContentPane;
    public Frame0402 焦點 1() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        JT1 = new JTextField("");
        JT2 = new JTextField("");
        JT3 = new JTextField("");
        JT4 = new JTextField("");
        JB1 = new JButton("TRANS");
        JT1.setBounds(130 · 60 · 90 · 28);

```




```
JT2.setBounds(230 · 148 · 90 · 28);
JT3.setBounds(130 · 232 · 90 · 28);
JT4.setBounds(30 · 148 · 90 · 28);
JB1.setBounds(130 · 148 · 90 · 28);
JT1.addFocusListener(new FocusAdapter() {
    public void focusGained(FocusEvent evt) {
        TextFocusGained(evt);
    }
});
JT1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        TextActionPerformed(evt);
    }
});
JT2.addFocusListener(new FocusAdapter() {
    public void focusGained(FocusEvent evt) {
        TextFocusGained(evt);
    }
});
JT2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        TextActionPerformed(evt);
    }
});
JT3.addFocusListener(new FocusAdapter() {
    public void focusGained(FocusEvent evt) {
        TextFocusGained(evt);
    }
});
JT3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        TextActionPerformed(evt);
    }
});
JT4.addFocusListener(new FocusAdapter() {
    public void focusGained(FocusEvent evt) {
        TextFocusGained(evt);
    }
});
JT4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        TextActionPerformed(evt);
    }
});
JB1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        JB1ActionPerformed(evt);
    }
});
ContentPane.add(JT1);
```

```
ContentPane.add(JT2);
ContentPane.add(JT3);
ContentPane.add(JT4);
ContentPane.add(JB1);
ContentPane.setBackground(Color.PINK);
this.setBounds(200 · 100 · 360 · 360);
this.setTitle(getClass().getName());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

public void TextFocusGained(FocusEvent evt) {
    JTextField tmp = (JTextField) evt.getSource();
    if (tmp == JT1) {
        JT1.selectAll();
    }
    if (tmp == JT2) {
        JT2.selectAll();
    }
    if (tmp == JT3) {
        JT3.selectAll();
    }
    if (tmp == JT4) {
        JT4.selectAll();
    }
}

public void TextActionPerformed(ActionEvent evt) {
    JTextField tmp = (JTextField) evt.getSource();
    if (tmp == JT1) {
        JT2.requestFocus();
    }
    if (tmp == JT2) {
        JT3.requestFocus();
    }
    if (tmp == JT3) {
        JT4.requestFocus();
    }
    if (tmp == JT4) {
        JT1.requestFocus();
    }
}

public void JB1ActionPerformed(ActionEvent evt) {
    String up · right · bottom · left;
    up = JT1.getText();
    right = JT2.getText();
    bottom = JT3.getText();
    left = JT4.getText();
    JT1.setText(left);
    JT2.setText(up);
    JT3.setText(right);
    JT4.setText(bottom);
}
```

```

    }

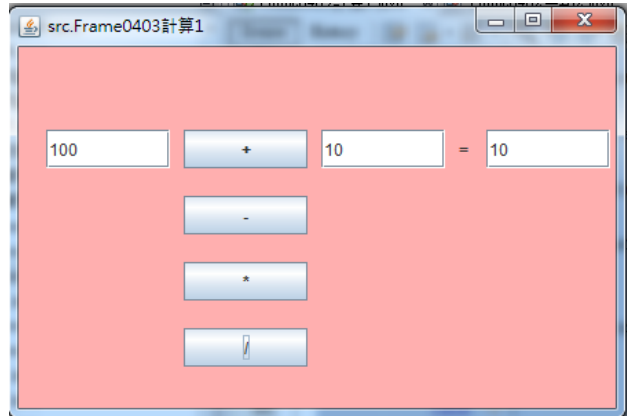
    public static void main(String[] args) {
        new Frame0402 焦點 1();
    }
}

```

```

public class Frame0403 計算 1 extends JFrame {
    public JTextField JT1 · JT2 · JT3;
    public JButton JB1 · JB2 · JB3 · JB4;
    public JLabel JL1;
    public Container ContentPane;
    public Frame0403 計算 1() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        JT1 = new JTextField("");
        JT2 = new JTextField("");
        JT3 = new JTextField("");
        JL1 = new JLabel("=");
        JB1 = new JButton("+");
        JB2 = new JButton("-");
        JB3 = new JButton("*");
        JB4 = new JButton("/");
        JT1.setBounds(20 · 60 · 90 · 28);
        JT2.setBounds(220 · 60 · 90 · 28);
        JT3.setBounds(340 · 60 · 90 · 28);
        JL1.setBounds(320 · 60 · 20 · 28);
        JB1.setBounds(120 · 60 · 90 · 28);
        JB2.setBounds(120 · 108 · 90 · 28);
        JB3.setBounds(120 · 156 · 90 · 28);
        JB4.setBounds(120 · 204 · 90 · 28);
        JB1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                count(evt);
            }
        });
        JB2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                count(evt);
            }
        });
        JB3.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                count(evt);
            }
        });
        JB4.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                count(evt);
            }
        });
    }
}

```



```

    }
    });
    ContentPane.add(JT1);
    ContentPane.add(JT2);
    ContentPane.add(JL1);
    ContentPane.add(JT3);
    ContentPane.add(JB1);
    ContentPane.add(JB2);
    ContentPane.add(JB3);
    ContentPane.add(JB4);
    ContentPane.setBackground(Color.PINK);
    this.setBounds(200, 100, 450, 300);
    this.setTitle(getClass().getName());
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setVisible(true);
}

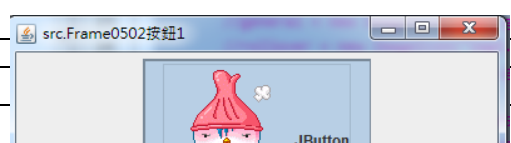
public void count(ActionEvent evt) {
    JButton tmp = (JButton) evt.getSource();
    int x = 0;
    String opt = tmp.getText();
    x = 四則運算(opt);
    JT3.setText(String.valueOf(x));
}

public int 四則運算(String op) {
    int x = 0;
    switch (op) {
        case "+":
            x = Integer.parseInt(JT1.getText()) + Integer.parseInt(JT2.getText());
            break;
        case "-":
            x = Integer.parseInt(JT1.getText()) - Integer.parseInt(JT2.getText());
            break;
        case "*":
            x = Integer.parseInt(JT1.getText()) * Integer.parseInt(JT2.getText());
            break;
        case "/":
            x = Integer.parseInt(JT1.getText()) / Integer.parseInt(JT2.getText());
            break;
    }
    return x;
}

public static void main(String[] args) {
    new Frame0403 計算 1();
}
}

```

```
public class Frame0501 按鈕 1 extends JFrame {
```



```
public Container ContentPane;
public JButton JB1;
public ImageIcon general;
public ImageIcon rollover;
public ImageIcon pressed;
public Frame0501 按鈕 1() {

    initComponents();
//放到 專案 的根目錄
    //general = new ImageIcon("img4.gif");
    //rollover = new ImageIcon("img5.gif");
    //pressed = new ImageIcon("img6.gif");
    //放到 src 的根目錄
    //general = new ImageIcon(Frame0502 按鈕 1.class.getResource("/pic/img4.gif"));
    //rollover = new ImageIcon(Frame0502 按鈕 1.class.getResource("/pic/img5.gif"));
    // pressed = new ImageIcon(Frame0502 按鈕 1.class.getResource("/pic/img6.gif"));
    general = new ImageIcon(getClass().getResource("/pic/img4.gif"));
    rollover = new ImageIcon(getClass().getResource("/pic/img5.gif"));
    pressed = new ImageIcon(getClass().getResource("/pic/img6.gif"));

    JB1.setRolloverEnabled(true);
    JB1.setIcon(general);
    JB1.setRolloverIcon(rollover);
    JB1.setPressedIcon(pressed);
}
private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(new FlowLayout());
    JB1 = new JButton("JButton");
    ContentPane.add(JB1);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setTitle(getClass().getName().toString());
    this.setBounds(300 · 300 · 400 · 300);
    this.setVisible(true);
}
public static void main(String[] args) {
    new Frame0501 按鈕 1();
}
}
```

```
public class Frame0502 表格 1 extends JFrame {
```

```
    public.JTable table1;
    public.JScrollPane scrollPane1;
    public.Container ContentPane;
    public String[] colname = {"名稱", "價格", "日期"};
    public String[][] data = {
        {"汽車", "1200", "10-01"},
        {"卡車", "2400", "10-05"},
        {"戰車", "3600", "10-06"},
        {"計程車", "2500", "10-10"},
        {"跑車", "2600", "10-11"},
        {"迷你車", "300", "10-12"},
        {"腳踏車", "800", "10-15"},
        {"三輪車", "600", "10-18"},
        {"飛機", "15000", "10-19"},
        {"直升機", "3500", "10-21"},
        {"火箭", "32800", "10-22"}
    };
};
```

```
public Frame0502 表格 1() {
```

```
    initComponents();
```

```
}
```

```
private void initComponents() {
```

```
    ContentPane = this.getContentPane();
```

```
    ContentPane.setLayout(null);
```

```
    table1 = new.JTable(data, colname);
```

```
    scrollPane1 = new.JScrollPane(table1);
```

```
    scrollPane1.setBounds(0, 30, 300, 200);
```

```
    ContentPane.add(scrollPane1);
```

```
    this.setTitle(getClass().getName());
```

```
    this.setBounds(200, 100, 600, 400);
```

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

```
    this.setVisible(true);
```

```
}
```

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

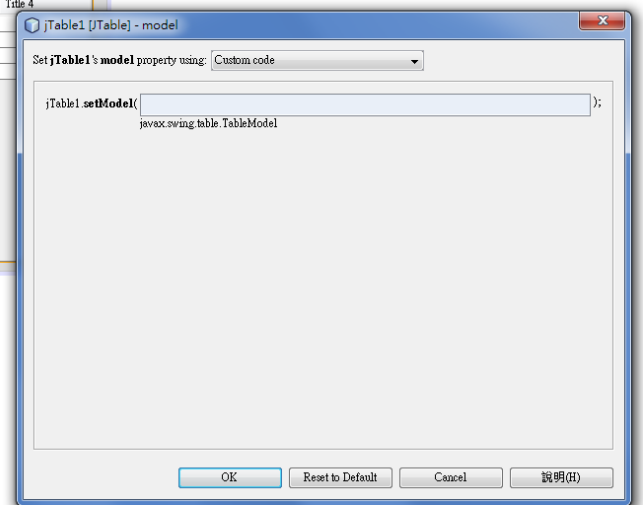
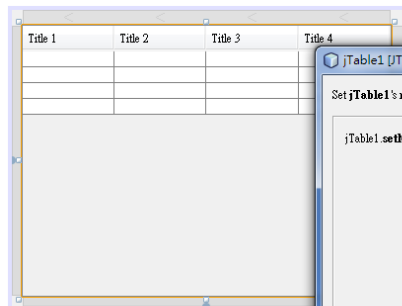
```
    new Frame0502 表格 1();
```

```
}
```

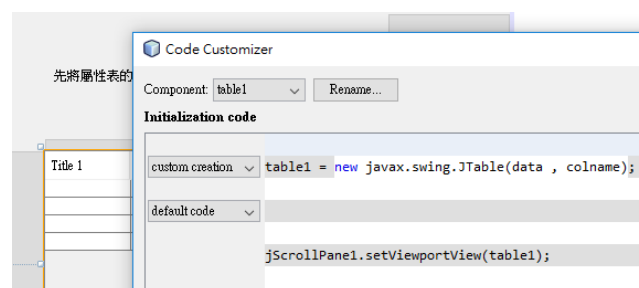
```
}
```



名稱	價格	日期
汽車	1200	10-01
卡車	2400	10-05
戰車	3600	10-06
計程車	2500	10-10
跑車	2600	10-11
迷你車	300	10-12
腳踏車	800	10-15
三輪車	600	10-18
飛機	15000	10-19
直升機	3500	10-21
火箭	32800	10-22

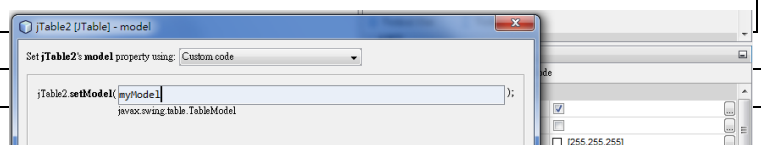


1.進入屬性表的 Model 選到 Custom code 按 ok 就能把 Model 模式取消，使用陣列



2.在畫面的表格上按右鍵把陣列指定進去

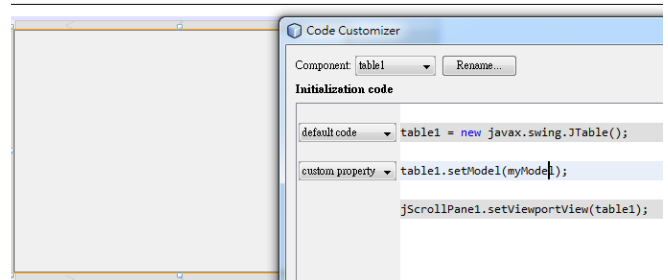
```
public class Frame0502 表格 2 extends JFrame {
```



```

public JScrollPane scrollPane1;
public Container ContentPane;
public JTable table1;
public MyTableModel myModel = new MyTableModel();
public Frame0502 表格 2() {
    initComponents();
}
private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(null);
    table1 = new JTable(myModel);
    scrollPane1 = new JScrollPane(table1);
    scrollPane1.setBounds(0, 30, 300, 200);
    ContentPane.add(scrollPane1);
    this.setTitle(getClass().getName());
    this.setBounds(200, 100, 700, 400);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setVisible(true);
}

```



// 模型類別

```

public class MyTableModel extends AbstractTableModel {
    String[] colname = {"名稱", "價格", "日期"};
    Object[][] data = {
        {"汽車", "1200", "10-01"},
        {"卡車", "2400", "10-05"},
        {"戰車", "3600", "10-06"},
        {"計程車", "2500", "10-10"},
        {"跑車", "2600", "10-11"},
        {"迷你車", "300", "10-12"},
        {"腳踏車", "800", "10-15"},
        {"三輪車", "600", "10-18"},
        {"飛機", "15000", "10-19"},
        {"直升機", "3500", "10-21"},
        {"火箭", "32800", "10-22"}
    };
    private int numColumns = colname.length;
    private int numRows = data.length;
    public MyTableModel() {
    }
    public int getRowCount() {
        return numRows;
    }
    public int getColumnCount() {
        return numColumns;
    }
    public Object getValueAt(int row, int column) {
        return data[row][column];
    }
    public String getColumnName(int column) {
        return colname[column];
    }
}

```

```

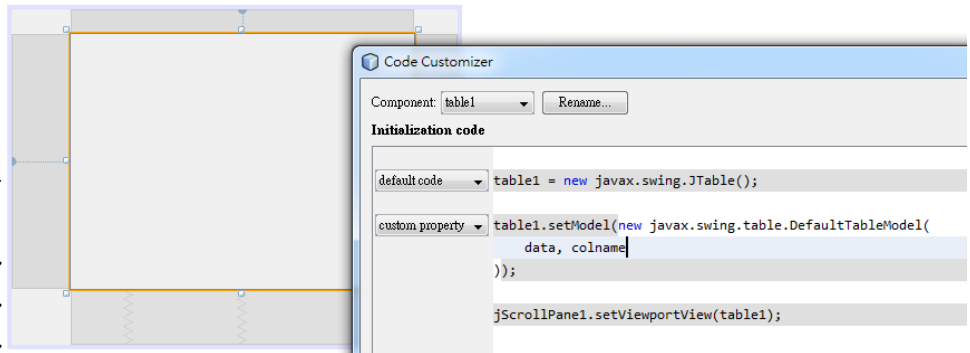
    }
    public boolean isCellEditable(int row · int column) {
        return true;
    }
    public void setValueAt(Object value · int row · int column) {
        data[row][column] = (String) value;
        fireTableCellUpdated(row · column);
    }
}
public static void main(String[] args) {
    new Frame0502 表格 2();
}
}

```

```

public class Frame0510 表格 3 extends JFrame {
    public JScrollPane scrollPane;
    public JTable table1;
    public Container ContentPane;
    public String[] colname = {"名稱", "價格", "日期"};
    public Object[][] data = {
        {"汽車", "1200", "10-01"},
        {"卡車", "2400", "10-05"},
        {"戰車", "3600", "10-06"},
        {"計程車", "2500", "10-10"},
        {"跑車", "2600", "10-11"},
        {"迷你車", "300", "10-12"},
        {"腳踏車", "800", "10-15"},
        {"三輪車", "600", "10-18"},
        {"飛機", "15000", "10-19"},
        {"直升機", "3500", "10-21"},
        {"火箭", "32800", "10-22"}
    };
    public Frame0502 表格 3() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        table1 = new JTable();
        table1.setModel(new javax.swing.table.DefaultTableModel(data · colname));
        scrollPane = new JScrollPane(table1);
        scrollPane.setBounds(0 · 30 · 300 · 200);
        ContentPane.add(scrollPane);
        this.setTitle(getClass().getName());
        this.setBounds(200 · 100 · 700 · 400);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
    public static void main(String[] args) {
        new Frame0502 表格 3();
    }
}

```

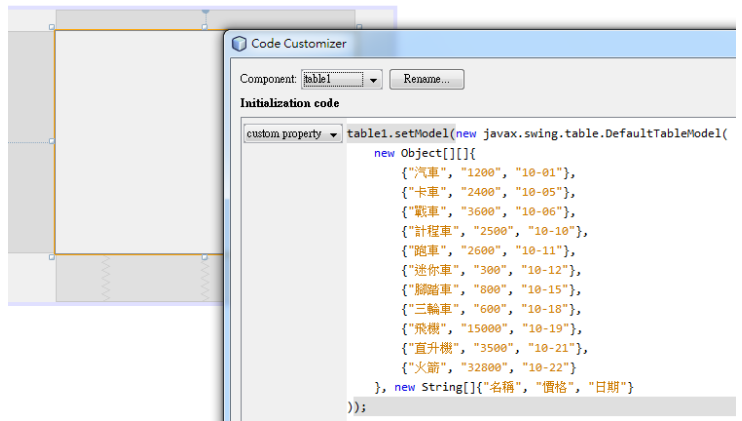



```

}

public class Frame0502 表格 4 extends JFrame {
    public JScrollPane scrollPane;
    public Container ContentPane;
    public JTable table1;
    public Frame0510 表格 4() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        table1 = new JTable();
        table1.setModel(new javax.swing.table.DefaultTableModel(
            new Object[][]{
                {"汽車", "1200", "10-01"},
                {"卡車", "2400", "10-05"},
                {"戰車", "3600", "10-06"},
                {"計程車", "2500", "10-10"},
                {"跑車", "2600", "10-11"},
                {"迷你車", "300", "10-12"},
                {"腳踏車", "800", "10-15"},
                {"三輪車", "600", "10-18"},
                {"飛機", "15000", "10-19"},
                {"直升機", "3500", "10-21"},
                {"火箭", "32800", "10-22"}
            },
            new String[]{"名稱", "價格", "日期"}));
        scrollPane = new JScrollPane(table1);
        scrollPane.setBounds(0, 30, 300, 200);
        ContentPane.add(scrollPane);
        this.setTitle(getClass().getName());
        this.setBounds(200, 100, 700, 400);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
    public static void main(String[] args) {
        new Frame0502 表格 4();
    }
}

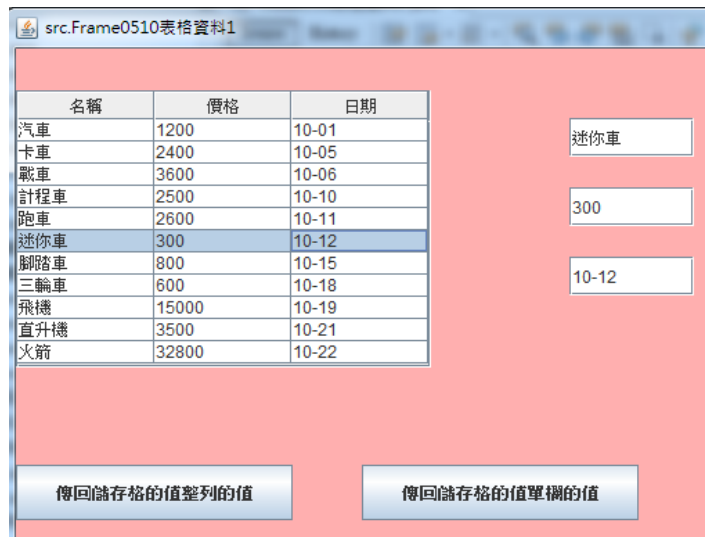
```



```

public class Frame0502 表格資料 1 extends JFrame {
    public JTextField JT1 · JT2 · JT3;
    public JTextField[] JT;
    public JButton JB1 · JB2;
    public JScrollPane scrollPane;
    public DefaultTableModel myModel;
    public Container ContentPane;
    public JTable table1;
    public Frame0510 表格資料 1() {
        initComponents();
        myModel = (DefaultTableModel) table1.getModel();
        JT = new JTextField[]{JT1 · JT2 · JT3};
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        table1 = new JTable();
        table1.setModel(new javax.swing.table.DefaultTableModel(
            new Object[][]{
                {"汽車", "1200", "10-01"},
                {"卡車", "2400", "10-05"},
                {"戰車", "3600", "10-06"},
                {"計程車", "2500", "10-10"},
                {"跑車", "2600", "10-11"},
                {"迷你車", "300", "10-12"},
                {"腳踏車", "800", "10-15"},
                {"三輪車", "600", "10-18"},
                {"飛機", "15000", "10-19"},
                {"直升機", "3500", "10-21"},
                {"火箭", "32800", "10-22"}
            },
            new String[]{"名稱", "價格", "日期"}));
        scrollPane = new JScrollPane(table1);
        JT1 = new JTextField("");
        JT2 = new JTextField("");
        JT3 = new JTextField("");
        JB1 = new JButton("傳回儲存格的值整列的值");
        JB2 = new JButton("傳回儲存格的值單欄的值");
        table1.addMouseListener(new MouseAdapter() {
            public void mouseClicked(MouseEvent evt) {
                choose1(evt);
            }
        });
        scrollPane.setBounds(0 · 30 · 300 · 200);
        JT1.setBounds(400 · 50 · 90 · 28);
        JT2.setBounds(400 · 100 · 90 · 28);
        JT3.setBounds(400 · 150 · 90 · 28);
        JB1.setBounds(0 · 300 · 200 · 40);
        JB2.setBounds(250 · 300 · 200 · 40);
        JB1.addActionListener(new ActionListener() {

```



```
        public void actionPerformed(ActionEvent evt) {
            choose2(evt);
        }
    });
    JB2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            choose3(evt);
        }
    });
    ContentPane.add(scrollPane);
    ContentPane.add(JT1);
    ContentPane.add(JT2);
    ContentPane.add(JT3);
    ContentPane.add(JB1);
    ContentPane.add(JB2);
    this.setTitle(getClass().getName());
    this.setBounds(200 · 100 · 700 · 400);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setVisible(true);
}

public void choose1(MouseEvent evt) {
    int selRow = table1.getSelectedRow();
    for (int i = 0; i < JT.length; i++) {
        JT[i].setText(myModel.getValueAt(selRow · i).toString());
    }
}

public void choose2(ActionEvent evt) {
    StringBuilder ss = new StringBuilder();
    String str;
    int selRow = table1.getSelectedRow();
    int selColumn = table1.getSelectedColumn();
    if (selRow == -1) {
        return;
    }
    for (int i = 0; i < table1.getColumnCount(); i++) {
        str = (String) myModel.getValueAt(selRow · i);
        if (i < table1.getColumnCount() - 1) {
            ss.append(str).append(" · ");
        } else {
            ss.append(str);
        }
    }
    JOptionPane.showMessageDialog(null · ss);
}

public void choose3(java.awt.event.ActionEvent evt) {
    StringBuilder ss = new StringBuilder();
    String str;
    int selRow = table1.getSelectedRow();
    int selColumn = table1.getSelectedColumn();
    if (selRow == -1) {
```

```

        return;
    }
    str = (String) myModel.getValueAt(selRow · selColumn);
    JOptionPane.showMessageDialog(null · str + " 在 " + selRow + "列" + selColumn + "欄");
}
public static void main(String[] args) {
    new Frame0502 表格資料 1();
}
}

```

Table 新增修改刪除

```
public class Frame0502 表格資料修改 1 extends JFrame {
```

```

    public JLabel JL1 · JL2 · JL3;
    public JTextField JT1 · JT2 · JT3;
    public JTextField[] JT;
    public JButton JB1 · JB2 · JB3 · JB4;
    public JScrollPane scrollPane;
    public DefaultTableModel myModel;
    public Container ContentPane;
    public JTable table1;
    public Frame0510 表格資料修改 1() {

```



```

        initComponents();
        myModel = (DefaultTableModel) table1.getModel();
        JT = new JTextField[]{JT1 · JT2 · JT3};
        for (JTextField elm : JT) {
            elm.setEditable(false);
        }
        //選取到第一列
        table1.setRowSelectionInterval(0 · 0);
        TableColumn colTitle = table1.getColumnModel().getColumn(0);
        colTitle.setPreferredWidth(100); //設定欄位的喜好寬度為 100
    }

    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        table1 = new JTable();
        table1.setModel(new javax.swing.table.DefaultTableModel(
            new Object[][]{
                {"汽車", "1200", "10-01"},
                {"卡車", "2400", "10-05"},
                {"戰車", "3600", "10-06"},
                {"計程車", "2500", "10-10"},
                {"跑車", "2600", "10-11"},
                {"迷你車", "300", "10-12"},
                {"腳踏車", "800", "10-15"},

```

```
        {"三輪車" , "600" , "10-18"} ,  
        {"飛機" , "15000" , "10-19"} ,  
        {"直升機" , "3500" , "10-21"} ,  
        {"火箭" , "32800" , "10-22"}  
    } ,  
    new String[]{"名稱" , "價格" , "日期"}));  
scrollPane = new JScrollPane(table1);  
JL1 = new JLabel("名稱");  
JL2 = new JLabel("價格");  
JL3 = new JLabel("日期");  
JT1 = new JTextField("");  
JT2 = new JTextField("");  
JT3 = new JTextField("");  
JB1 = new JButton("新增");  
JB2 = new JButton("修改");  
JB3 = new JButton("刪除");  
JB4 = new JButton("取消");  
JB4.setEnabled(false);  
scrollPane.setBounds(0 , 30 , 300 , 200);  
JL1.setBounds(320 , 35 , 90 , 28);  
JL2.setBounds(320 , 70 , 90 , 28);  
JL3.setBounds(320 , 105 , 90 , 28);  
JT1.setBounds(400 , 35 , 90 , 28);  
JT2.setBounds(400 , 70 , 90 , 28);  
JT3.setBounds(400 , 105 , 90 , 28);  
JB1.setBounds(300 , 250 , 90 , 28);  
JB2.setBounds(390 , 250 , 90 , 28);  
JB3.setBounds(480 , 250 , 90 , 28);  
JB4.setBounds(570 , 250 , 90 , 28);  
table1.addMouseListener(new MouseAdapter() {  
    public void mouseClicked(MouseEvent evt) {  
        choose1(evt);  
    }  
});  
JB1.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        新增(evt);  
    }  
});  
JB2.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        修改(evt);  
    }  
});
```

```
});
JB3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        刪除(evt);
    }
});
JB4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        清除選取(evt);
    }
});
ContentPane.add(scrollPane);
ContentPane.add(JL1);
ContentPane.add(JL2);
ContentPane.add(JL3);
ContentPane.add(JT1);
ContentPane.add(JT2);
ContentPane.add(JT3);
ContentPane.add(JB1);
ContentPane.add(JB2);
ContentPane.add(JB3);
ContentPane.add(JB4);
this.setTitle(getClass().getName().toString());
this.setBounds(200 · 100 · 700 · 400);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
public void choose1(MouseEvent evt) {
    int selRow = table1.getSelectedRow();
    for (int i = 0; i < JT.length; i++) {
        JT[i].setText(myModel.getValueAt(selRow · i).toString());
    }
}
public void 新增(java.awt.event.ActionEvent evt) {
    if (evt.getActionCommand().equals("新增")) {
        for (JTextField elm : JT) {
            elm.setEditable(true);
            elm.setText(null);
        }
        JB1.setText("儲存");
        JB2.setEnabled(false);
        JB3.setEnabled(false);
        JB4.setEnabled(true);
    }
}
```

```
} else if (evt.getActionCommand().equals("儲存")) {  
    Object[] newRow = {  
        JT1.getText() ·  
        JT2.getText() ·  
        JT3.getText()  
    };  
    myModel.addRow(newRow);  
    int newRowIndex = myModel.getRowCount() - 1;  
    table1.setRowSelectionInterval(newRowIndex · newRowIndex);  
    for (JTextField elm : JT) {  
        elm.setEditable(false);  
    }  
    JB1.setText("新增");  
    JB2.setEnabled(true);  
    JB3.setEnabled(true);  
    JB4.setEnabled(false);  
}  
}  
  
public void 修改(java.awt.event.ActionEvent evt) {  
    if (evt.getActionCommand().equals("修改")) {  
        //判斷是否選取不止一列 · 是則設定僅選取第一列  
        if (table1.getSelectedRows().length != 1) {  
            int selRow = table1.getSelectedRow();  
            //取得選取範圍的第一列  
            table1.setRowSelectionInterval(selRow · selRow);  
        }  
        for (JTextField elm : JT) {  
            elm.setEditable(true);  
        }  
        JB2.setText("儲存");  
        JB1.setEnabled(false);  
        JB3.setEnabled(false);  
        JB4.setEnabled(true);  
    } else if (evt.getActionCommand().equals("儲存")) {  
        Object[] newData = {  
            JT1.getText() ·  
            JT2.getText() ·  
            JT3.getText()  
        };  
        int selRow = table1.getSelectedRow();  
        for (int i = 0; i < JT.length; i++) {  
            JT[i].setEditable(false);  
            myModel.setValueAt(newData[i] · selRow · i);  
        }  
    }  
}
```

```
    }
    JB2.setText("修改");
    JB1.setEnabled(true);
    JB3.setEnabled(true);
    JB4.setEnabled(false);
}
}

public void 刪除(java.awt.event.ActionEvent evt) {
    int[] selRows = table1.getSelectedRows();
    if (selRows.length == 0) {
        return;
    }
    //依照選取列的索引值移除資料
    for (int i = selRows.length - 1; i >= 0; i--) {
        myModel.removeRow(selRows[i]);
        //必須從後面開始移除，
        //因為移除一行資料時，後面選取列的索引值將改變
        //從最後一行開始移除則可解決此問題
    }
    int rowCount = myModel.getRowCount();
    if (rowCount == 0) {
        for (JTextField elm : JT) {
            elm.setText(null);
        }
        return;
    }
    //刪除後索引值選第一列
    table1.setRowSelectionInterval(0, 0);
    int selRow = table1.getSelectedRow();
    for (int i = 0; i < JT.length; i++) {
        JT[i].setText(table1.getValueAt(selRow, i).toString());
    }
}

public void 清除選取(java.awt.event.ActionEvent evt) {
    int selRow = table1.getSelectedRow();
    table1.clearSelection(); //清除選取
    table1.setRowSelectionInterval(selRow, selRow); //重設選取範圍
    for (JTextField elm : JT) {
        elm.setEditable(false); //將文字欄設定為不可編輯
    }
    JB4.setEnabled(false);
    //將取消按鈕設定為無效
    JB1.setText("新增");
}
```



```

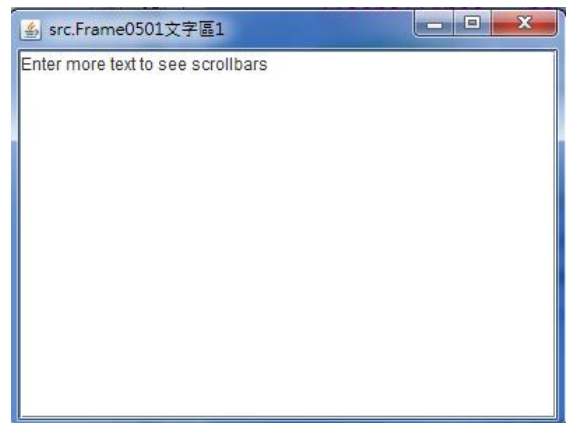
        JB2.setText("修改");
        JB1.setEnabled(true);
        JB2.setEnabled(true);
        JB3.setEnabled(true);
    }
    public static void main(String[] args) {
        new Frame0502 表格資料修改 1();
    }
}

```

```

public class Frame0503 文字區 1 extends JFrame {
    JTextArea resultArea;
    JScrollPane scrollArea;
    Container ContentPane;
    public Frame0503 文字區 1() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(new BorderLayout());
        // resultArea = new JTextArea(6 * 20);
        resultArea = new JTextArea();
        resultArea.setColumns(20);
        resultArea.setRows(6);
        resultArea.setText("Enter more text to see scrollbars\n");
        //scrollArea = new JScrollPane(resultArea);
        scrollArea = new JScrollPane();
        scrollArea.setViewportView(resultArea);
        ContentPane.add(scrollArea, BorderLayout.CENTER);
        this.setTitle(getClass().getName());
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setBounds(300, 300, 400, 300);
        this.setVisible(true);
    }
    public static void main(String[] args) {
        new Frame0503 文字區 1();
    }
}

```



```

public class Frame0504 標籤 1 extends JFrame {

```



```
public ImageIcon pic[];
public ImageIcon imgleft;
public ImageIcon imgright;
public JButton JB1;
public JButton JB2;
public JLabel JL1;
public Container ContentPane;
public int index = 0;
public Frame0504 標籤 1() {
    pic = new ImageIcon[4];
    pic[0] = new ImageIcon(getClass()
        .getResource("/pic/sp0.jpg"));
    pic[1] = new ImageIcon(getClass()
        .getResource("/pic/sp1.jpg"));
    pic[2] = new ImageIcon(getClass()
        .getResource("/pic/sp2.jpg"));
    pic[3] = new ImageIcon(getClass()
        .getResource("/pic/sp3.jpg"));
    imgleft = new ImageIcon(getClass()
        .getResource("/pic/left.gif"));
    imgright = new ImageIcon(getClass()
        .getResource("/pic/right.gif"));
    initComponents();
    JB2.setHorizontalTextPosition(JButton.LEFT);
    JL1.setIcon(pic[0]);
    JL1.setText("sp0.jpg");
    JL1.setHorizontalAlignment(JLabel.CENTER);
    JL1.setVerticalAlignment(JLabel.BOTTOM);
}
private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(new FlowLayout());
    JB1 = new JButton("前一張" · imgleft);
    JB2 = new JButton("後一張" · imgright);
    JL1 = new JLabel();
    JB1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            showpic(evt);
        }
    });
    JB2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            showpic(evt);
        }
    });
}
```

```

    }

    });

    ContentPane.add(JB1);
    ContentPane.add(JB2);
    ContentPane.add(JL1);
    this.setTitle(getClass().getName());
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setSize(400 · 350);
    this.setVisible(true);
}

public void showpic(ActionEvent evt) {
    JButton tmp = (JButton) evt.getSource();
    int num = pic.length;
    if (tmp == JB1 && index >= 0) {
        if (index == 0) {
            index = num;
        }
        index--;
    }
    if (tmp == JB2 && index <= num - 1) {
        if (index == num - 1) {
            index = -1;
        }
        index++;
    }
    JL1.setText("sp" + (index) + ".jpg");
    JL1.setIcon(pic[index]);
}

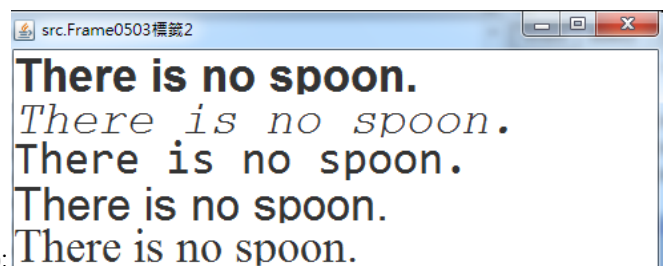
public static void main(String[] args) {
    new Frame0504 標籤 1();
}
}

```

```

public class Frame0504 標籤 2 extends JFrame {
    public Container ContentPane;
    public JLabel word1 · word2 · word3 · word4 · word5;
    public Frame0504 標籤 2() {
        initComponents();
        Font font1 = new Font(Font.DIALOG · Font.BOLD · 35);
        word1.setFont(font1);
        Font font2 = new Font(Font.DIALOG_INPUT · Font.ITALIC · 35);
        word2.setFont(font2);
        Font font3 = new Font(Font.MONOSPACED · Font.PLAIN · 35);
        word3.setFont(font3);

```



```
Font font4 = new Font(Font.SANS_SERIF · Font.PLAIN · 35);
word4.setFont(font4);

Font font5 = new Font(Font.SERIF · Font.PLAIN · 35);
word5.setFont(font5);

}

private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(new GridLayout(5 · 1));
    word1 = new JLabel("There is no spoon.");
    word2 = new JLabel("There is no spoon.");
    word3 = new JLabel("There is no spoon.");
    word4 = new JLabel("There is no spoon.");
    word5 = new JLabel("There is no spoon.");
    ContentPane.add(word1);
    ContentPane.add(word2);
    ContentPane.add(word3);
    ContentPane.add(word4);
    ContentPane.add(word5);
    ContentPane.setBackground(Color.WHITE);
    this.setTitle(getClass().getName());
    this.setBounds(200 · 100 · 500 · 200);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setVisible(true);
}

public static void main(String[] args) {
    new Frame0504 標籤 2();
}

}
```

核取框 → itemStateChanged 事件

```
public class Frame0505 核取框 1 extends JFrame {
    public JLabel JL1;
    public JCheckBox JCH1 · JCH2;
    public JPanel JP1;
    public Container ContentPane;
    public Frame0505 核取框 1() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        GridLayout layout = new GridLayout(1 · 0);
        layout.setHgap(15);
        JL1 = new JLabel();
```

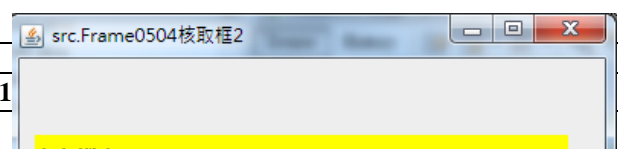


```
JP1 = new JPanel();
JCH1 = new JCheckBox("汽車");
JCH2 = new JCheckBox("卡車");
JL1.setBounds(10 · 50 · 300 · 40);
JL1.setBackground(Color.YELLOW);
JL1.setOpaque(true);
JP1.setBounds(10 · 100 · 300 · 40);
JCH1.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose(evt);
    }
});
JCH2.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose(evt);
    }
});
JP1.setLayout(layout);
JP1.add(JCH1);
JP1.add(JCH2);
ContentPane.add(JL1);
ContentPane.add(JP1);
this.setTitle(getClass().getName());
this.setBounds(200 · 100 · 350 · 250);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

public void choose(ItemEvent evt) {
    JCheckBox tmp = (JCheckBox) evt.getSource();
    if (evt.getStateChange() == ItemEvent.SELECTED) {
        JL1.setText("選擇了" + tmp.getText() + "。");
    } else if (evt.getStateChange() == ItemEvent.DESELECTED) {
        JL1.setText("放棄了" + tmp.getText() + "。");
    }
}

public static void main(String[] args) {
    new Frame0505 核取框 1();
}
}
```

```
public class Frame0505 核取框 2 extends JFrame {
```



```
public JCheckBox JCH1;
public JCheckBox JCH2;
public JCheckBox JCH3;
public JCheckBox JCH4;
public JLabel JL1;
public JPanel JP1;
public Container ContentPane;
public JCheckBox[] JCHK;
public Frame0505 核取框 2() {
    initComponents();
    JCHK = new javax.swing.JCheckBox[]{JCH1 · JCH2 · JCH3 · JCH4};
}
private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(null);
    GridLayout layout = new GridLayout(1 · 0);
    layout.setHgap(15);
    JL1 = new JLabel();
    JP1 = new JPanel();
    JCH1 = new JCheckBox("汽車");
    JCH2 = new JCheckBox("卡車");
    JCH3 = new JCheckBox("機車");
    JCH4 = new JCheckBox("腳踏車");
    JL1.setBounds(10 · 50 · 350 · 28);
    JL1.setBackground(Color.yellow);
    JL1.setOpaque(true);
    JP1.setBounds(10 · 100 · 350 · 28);
    JCH1.addItemListener(new java.awt.event.ItemListener() {
        public void itemStateChanged(java.awt.event.ItemEvent evt) {
            choose(evt);
        }
    });
    JCH2.addItemListener(new java.awt.event.ItemListener() {
        public void itemStateChanged(java.awt.event.ItemEvent evt) {
            choose(evt);
        }
    });
    JCH3.addItemListener(new java.awt.event.ItemListener() {
        public void itemStateChanged(java.awt.event.ItemEvent evt) {
            choose(evt);
        }
    });
    JCH4.addItemListener(new java.awt.event.ItemListener() {
        public void itemStateChanged(java.awt.event.ItemEvent evt) {
```

```

        choose(evt);
    }
});
JP1.setLayout(layout);
JP1.add(JCH1);
JP1.add(JCH2);
JP1.add(JCH3);
JP1.add(JCH4);
ContentPane.add(JL1);
ContentPane.add(JP1);
this.setTitle(getClass().getName());
this.setBounds(200 · 100 · 400 · 250);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

public void choose(java.awt.event.ItemEvent evt) {
    StringBuilder str = new StringBuilder("");
    for (int i = 0; i < JCHK.length; i++) {
        if (JCHK[i].isSelected()) {
            str.append(JCHK[i].getText()).append(" ");
        }
    }
    JL1.setText(str.toString());
}

public static void main(String args[]) {
    new Frame0505 核取框 2();
}
}

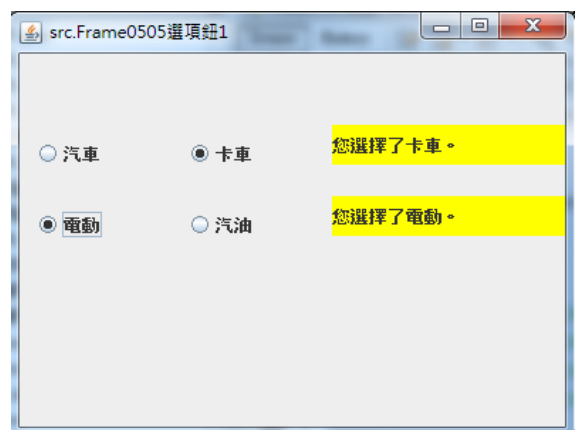
```

選項鈕 → ItemEvent 事件 · Action 事件

```

public class Frame0506 選項鈕 1 extends JFrame {
    public JPanel JP1 · JP2;
    public JLabel JL1 · JL2;
    public ButtonGroup buttongroup1 · buttongroup2;
    public JRadioButton JRB1 · JRB2 · JRB3 · JRB4;
    public Container ContentPane;
    public Frame0506 選項鈕 1() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        GridLayout layout = new GridLayout(1 · 0);
        layout.setHgap(15);
    }
}

```



```
JL1 = new JLabel();
JL2 = new JLabel();
JP1 = new JPanel();
JP2 = new JPanel();
JRB1 = new JRadioButton("汽車" , true);
JRB2 = new JRadioButton("卡車" , false);
JRB3 = new JRadioButton("電動" , true);
JRB4 = new JRadioButton("汽油" , false);
buttongroup1 = new ButtonGroup();
buttongroup2 = new ButtonGroup();
JL1.setBounds(220 , 50 , 200 , 28);
JL1.setBackground(Color.yellow);
JL1.setOpaque(true);
JL2.setBounds(220 , 100 , 200 , 28);
JL2.setBackground(Color.yellow);
JL2.setOpaque(true);
JP1.setBounds(10 , 50 , 200 , 40);
JP2.setBounds(10 , 100 , 200 , 40);
JRB1.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose1(evt);
    }
});
JRB2.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose1(evt);
    }
});
JRB3.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose2(evt);
    }
});
JRB4.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose2(evt);
    }
});
buttongroup1.add(JRB1);
buttongroup1.add(JRB2);
buttongroup2.add(JRB3);
buttongroup2.add(JRB4);
JP1.setLayout(layout);
```



```
JP1.add(JRB1);
JP1.add(JRB2);
JP2.setLayout(layout);
JP2.add(JRB3);
JP2.add(JRB4);
ContentPane.add(JP1);
ContentPane.add(JP2);
ContentPane.add(JL1);
ContentPane.add(JL2);
this.setTitle(getClass().getName());
this.setBounds(200 · 100 · 400 · 300);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

private void choose1(java.awt.event.ItemEvent evt) {
    JRadioButton tmp = (JRadioButton) evt.getSource();
    JL1.setText("您選擇了" + tmp.getText() + " ° ");
}

public void choose2(java.awt.event.ItemEvent evt) {
    JRadioButton tmp = (JRadioButton) evt.getSource();
    JL2.setText("您選擇了" + tmp.getText() + " ° ");
}

public static void main(String[] args) {
    new Frame0506 選項鈕 1();
}
}
```

```
public class Frame0506 選項鈕 2 extends JFrame {
    public JPanel JP1 · JP2;
    public JLabel JL1 · JL2;
    public ButtonGroup buttongroup1 · buttongroup2;
    public JRadioButton JRB1 · JRB2 · JRB3 · JRB4;
    public Container ContentPane;
    public JRadioButton[] JRBX · JRBY;
    public Frame0506 選項鈕 2() {
        initComponents();
        JRBX = new JRadioButton[]{JRB1 · JRB2};
        JRBY = new JRadioButton[]{JRB3 · JRB4};
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        GridLayout layout = new GridLayout(1 · 0);
        layout.setHgap(15);
        JL1 = new JLabel();
        JL2 = new JLabel();
        JP1 = new JPanel();
```

```
JP2 = new JPanel();
JRB1 = new JRadioButton("汽車" · true);
JRB2 = new JRadioButton("卡車" · false);
JRB3 = new JRadioButton("電動" · true);
JRB4 = new JRadioButton("汽油" · false);
buttongroup1 = new ButtonGroup();
buttongroup2 = new ButtonGroup();
JL1.setBounds(220 · 50 · 200 · 28);
JL1.setBackground(Color.yellow);
JL1.setOpaque(true);
JL2.setBounds(220 · 100 · 200 · 28);
JL2.setBackground(Color.yellow);
JL2.setOpaque(true);
JP1.setBounds(10 · 50 · 200 · 40);
JP2.setBounds(10 · 100 · 200 · 40);
JRB1.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose1(evt);
    }
});
JRB2.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose1(evt);
    }
});
JRB3.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose2(evt);
    }
});
JRB4.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose2(evt);
    }
});
buttongroup1.add(JRB1);
buttongroup1.add(JRB2);
buttongroup2.add(JRB3);
buttongroup2.add(JRB4);
JP1.setLayout(layout);
JP1.add(JRB1);
JP1.add(JRB2);
JP2.setLayout(layout);
JP2.add(JRB3);
JP2.add(JRB4);
ContentPane.add(JP1);
ContentPane.add(JP2);
ContentPane.add(JL1);
ContentPane.add(JL2);
this.setTitle(getClass().getName());
```

```

        this.setBounds(200 · 100 · 400 · 300);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
    private void choose1(java.awt.event.ItemEvent evt) {
        for (int i = 0; i < JRBX.length; i++) {
            if (JRBX[i].isSelected()) {
                JL1.setText("您選擇了" + JRBX[i].getText());
            }
        }
    }
    private void choose2(java.awt.event.ItemEvent evt) {
        for (int i = 0; i < JRBY.length; i++) {
            if (JRBY[i].isSelected()) {
                JL2.setText("您選擇了" + JRBY[i].getText());
            }
        }
    }
    public static void main(String[] args) {
        new Frame0506 選項鈕 2();
    }
}

```

```

public class Frame0507 選項鈕與核取框 1 extends JFrame {
    public JTextField JT1;
    public JCheckBox checkB · checkI;
    public JRadioButton radioL · radioC · radioR;
    public JComboBox comboBox;
    public ButtonGroup buttonGroup;
    public JPanel JP1 · JP2;
    public Container ContentPane;
    public int n = 2; //向左
    //int bold = Font.BOLD · italic = Font.ITALIC;
    public int bold = 0 · italic = 0;
    public int align = JTextField.LEFT · fonts = 14;
    public String name[] = {"14" · "16" · "18" · "20"};
    public Frame0507 選項鈕與核取框 1() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = getContentPane();
        ContentPane.setLayout(null);
        GridLayout layout = new GridLayout(1 · 0);
        layout.setHgap(15);
        JT1 = new JTextField("Just a font Tester");
        JT1.setFont(new Font("Serif" · Font.PLAIN · 14));
        radioL = new JRadioButton("向左");
        radioC = new JRadioButton("置中");
        radioR = new JRadioButton("向右");
        checkB = new JCheckBox("粗體" · false);
        checkI = new JCheckBox("斜體" · false);
    }
}

```



```
comboBox = new JComboBox(name);
comboBox.setMaximumRowCount(3);
buttonGroup = new ButtonGroup();
JP1 = new JPanel();
JP2 = new JPanel();
JT1.setBounds(30 · 10 · 250 · 28);
JP1.setBounds(30 · 50 · 250 · 28);
JP2.setBounds(30 · 90 · 250 · 28);
comboBox.setBounds(30 · 150 · 250 · 28);
checkB.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose(evt);
    }
});
checkI.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose(evt);
    }
});
radioL.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose(evt);
    }
});
radioC.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose(evt);
    }
});
radioR.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose(evt);
    }
});
comboBox.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose(evt);
    }
});
buttonGroup.add(radioL);
buttonGroup.add(radioC);
buttonGroup.add(radioR);
JP1.setLayout(layout);
JP1.add(radioL);
JP1.add(radioR);
JP1.add(radioC);
JP2.setLayout(layout);
JP2.add(checkB);
JP2.add(checkI);
ContentPane.add(JT1);
```

```

ContentPane.add(JP1);
ContentPane.add(JP2);
ContentPane.add(comboBox);
this.setTitle(getClass().getName());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setSize(350 · 300);
this.setVisible(true);
}

public void choose(ItemEvent evt) {
    //選項鈕 靠左 · 置中 · 靠右
    //int n = 2;
    if (evt.getSource() == radioL) {
        n = JTextField.LEFT;
    } else if (evt.getSource() == radioR) {
        n = JTextField.RIGHT;
    } else if (evt.getSource() == radioC) {
        n = JTextField.CENTER;
    }
    JT1.setHorizontalAlignment(n);
    //核取框 粗體 · 斜體
    if (evt.getSource() == checkB) {
        if (evt.getStateChange() == ItemEvent.SELECTED) {
            bold = 1;
        } else {
            bold = 0;
        }
    }
    if (evt.getSource() == checkI) {
        if (evt.getStateChange() == ItemEvent.SELECTED) {
            italic = 2;
        } else {
            italic = 0;
        }
    }
    //字型
    String big = "14";
    if (evt.getSource() == comboBox) {
        big = (String) comboBox.getSelectedItem();
    }
    JT1.setFont(new Font("Serif" · bold + italic · Integer.parseInt(big)));
}

public static void main(String[] args) {
    new Frame0507 選項鈕與核取框 1();
}
}

```

清單→ListSelectionListener 的 valueChanged 事件 · 使用陣列屬性表 model 改成 customer · 使用 model 屬性表 model 不改

```
public class Frame0508 清單 1 extends JFrame {
```

src.Frame0507清單1

您選擇了 2月 5月 8月

汽車

1月

```
public JLabel JL1;
public JList JLST1 , JLST2;
public JScrollPane scrollPane1 , scrollPane2;
public Container ContentPane;
public String[] traffic = {"汽車" , "卡車" , "戰車" ,
    "計程車" , "跑車" , "迷你車" ,
    "腳踏車" , "三輪車" , "摩托車" ,
    "飛機" , "直升機" , "火箭"};
public String[] months = {"1 月" , "2 月" , "3 月" , "4 月" , "5 月" , "6 月" , "7 月" , "8 月" , "9 月" ,
    "10 月" , "11 月" , "12 月"};
public Frame0508 清單 1() {
    initComponents();
}
private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(null);
    JL1 = new JLabel();
    JLST1 = new JList(traffic);
    JLST2 = new JList(months);
    scrollPane1 = new JScrollPane(JLST1);
    scrollPane2 = new JScrollPane(JLST2);
    JL1.setBackground(Color.YELLOW);
    JL1.setOpaque(true);
    JLST1.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);// 設置選擇模式
    JLST2.setSelectionMode(ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);// 設置選擇模式
    JLST2.setFixedCellHeight(20);                // 設置選項高度
    JLST2.setVisibleRowCount(6);                // 設置可見個數
    JLST2.setSelectionBackground(Color.GREEN);    // 設置被選中項的背景顏色
    JLST2.setSelectionForeground(Color.RED);      // 設置被選中項的字體顏色
    JL1.setBounds(20 , 0 , 400 , 40);
    scrollPane1.setBounds(20 , 50 , 150 , 200);
    scrollPane2.setBounds(200 , 50 , 150 , 200);
    JLST1.addListSelectionListener(new ListSelectionListener() {
        public void valueChanged(ListSelectionEvent evt) {
            choose(evt);
        }
    });
    JLST2.addListSelectionListener(new ListSelectionListener() {
        public void valueChanged(ListSelectionEvent evt) {
            choose(evt);
        }
    });
    ContentPane.add(JL1);
```

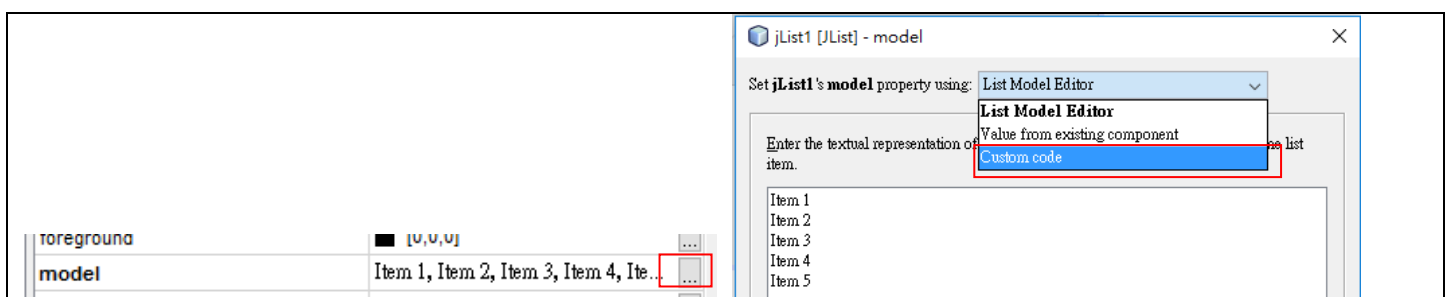
```

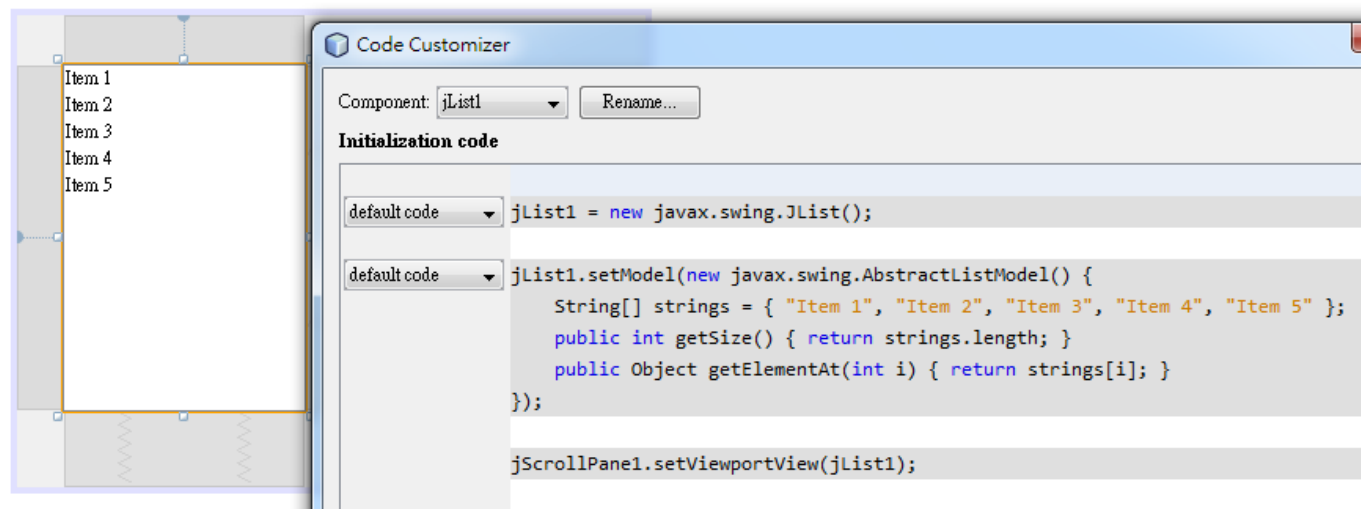
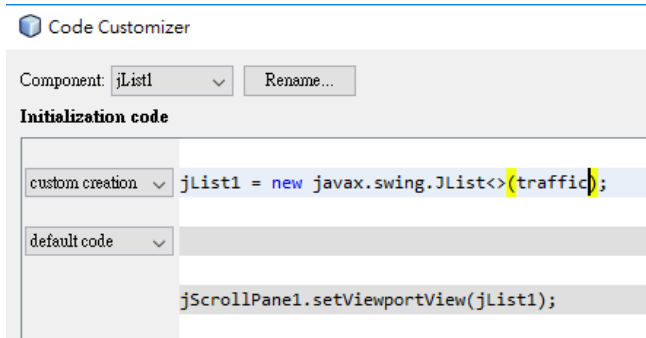
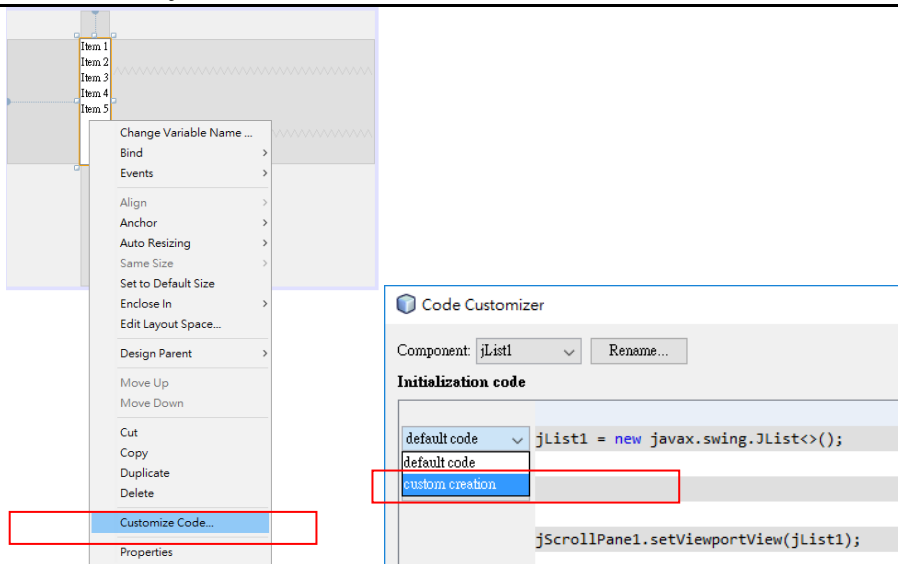
ContentPane.add(scrollPane1);
ContentPane.add(scrollPane2);
this.setTitle(getClass().getName());
this.setBounds(200 · 100 · 500 · 400);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

public void choose(ListSelectionEvent evt) {
    JL1.setText("您選擇了");
    JList tmp = (JList) evt.getSource();
    if (tmp == JLST1) {
        String str = (String) tmp.getSelectedValue();
        JL1.setText(JL1.getText() + "    " + str);
    }
    if (tmp == JLST2) {
        Object[] selectedValues = tmp.getSelectedValues();
        for (int i = 0; i < selectedValues.length; i++) {
            JL1.setText(JL1.getText() + "    " + selectedValues[i]);
        }
    }
}

public static void main(String[] args) {
    new Frame0508 清單 1();
}
}

```





```
public class Frame0508 清單 2 extends JFrame {
    public JLabel JL1;
    public JList JLST1 · JLST2;
    public JScrollPane scrollPane1 · scrollPane2;
    public Container ContentPane;
    public Frame0508 清單 2() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        JL1 = new JLabel();
```



```
JLST1 = new JList();
JLST2 = new JList();
scrollPane1 = new JScrollPane();
scrollPane2 = new JScrollPane();
JL1.setBackground(Color.YELLOW);
JL1.setOpaque(true);
JLST1.setModel(new AbstractListModel() {
    String[] strings = {"汽車", "卡車", "戰車",
        "計程車", "跑車", "迷你車",
        "腳踏車", "三輪車", "摩托車",
        "飛機", "直升機", "火箭"};
    public int getSize() {
        return strings.length;
    }
    public Object getElementAt(int i) {
        return strings[i];
    }
});
JLST1.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);// 設置選擇模式
JLST2.setModel(new javax.swing.AbstractListModel() {
    String[] strings = {"1 月", "2 月", "3 月", "4 月", "5 月", "6 月", "7 月", "8 月", "9 月", "10 月", "11
月", "12 月"};
    public int getSize() {
        return strings.length;
    }
    public Object getElementAt(int i) {
        return strings[i];
    }
});
JLST2.setSelectionMode(ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);// 設置選擇模式
JLST2.setFixedCellHeight(20);           // 設置選項高度
JLST2.setVisibleRowCount(6);           // 設置可見個數
JLST2.setSelectionBackground(Color.GREEN); // 設置被選中項的背景顏色
JLST2.setSelectionForeground(Color.RED);  // 設置被選中項的字體顏色
scrollPane1 = new JScrollPane();
scrollPane2 = new JScrollPane();
JL1.setBounds(20, 0, 400, 40);
scrollPane1.setBounds(20, 50, 150, 200);
scrollPane2.setBounds(200, 50, 150, 200);
JLST1.addListSelectionListener(new ListSelectionListener() {
    public void valueChanged(ListSelectionEvent evt) {
        choose(evt);
    }
});
```

```

});
scrollPane1.setViewportView(JLST1);
JLST2.addListSelectionListener(new ListSelectionListener() {
    public void valueChanged(ListSelectionEvent evt) {
        choose(evt);
    }
});
scrollPane2.setViewportView(JLST2);
ContentPane.add(JL1);
ContentPane.add(scrollPane1);
ContentPane.add(scrollPane2);
this.setTitle(getClass().getName());
this.setBounds(200 · 100 · 500 · 400);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

public void choose(ListSelectionEvent evt) {
    JL1.setText("您選擇了");
    JList tmp = (JList) evt.getSource();
    if (tmp == JLST1) {
        String str = (String) tmp.getSelectedValue();
        JL1.setText(JL1.getText() + "    " + str);
    }
    if (tmp == JLST2) {
        Object[] selectedValues = tmp.getSelectedValues();
        for (int i = 0; i < selectedValues.length; i++) {
            JL1.setText(JL1.getText() + "    " + selectedValues[i]);
        }
    }
}

public static void main(String[] args) {
    new Frame0508 清單 20);
}
}

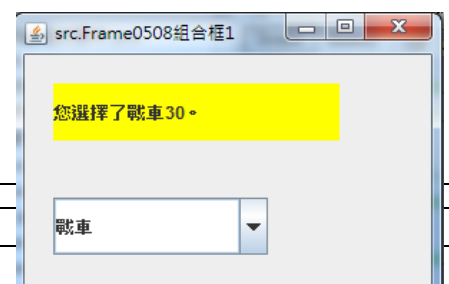
```

組合框→ItemEvent · Action 事件

```

public class Frame0509 組合框 1 extends JFrame {
    public JLabel JL1;
    public JComboBox JCB1;
    public Container ContentPane;
    public String[] traffic = {"汽車", "卡車", "戰車", "計程車", "跑車", "迷你車"};

```



```
public Frame0509 組合框 1() {
    initComponents();
}

private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(null);
    JL1 = new JLabel("");
    JCB1 = new JComboBox(traffic);
    JL1.setBackground(Color.yellow);
    JL1.setOpaque(true);
    JCB1.setEditable(true);
    JL1.setBounds(20 · 20 · 200 · 40);
    JCB1.setBounds(20 · 100 · 150 · 40);
    JCB1.addItemListener(new ItemListener() {
        public void itemStateChanged(ItemEvent evt) {
            choose(evt);
        }
    });
    ContentPane.add(JL1);
    ContentPane.add(JCB1);
    this.setTitle(getClass().getName());
    this.setBounds(200 · 100 · 300 · 250);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setVisible(true);
}

public void choose(ItemEvent evt) {
    int[] ar1 = {10 · 20 · 30 · 40 · 50 · 60};
    JComboBox tmp = (JComboBox) evt.getSource();
    String str = (String) tmp.getSelectedItem();
    JL1.setText("您選擇了" + str + ar1[tmp.getSelectedIndex()] + " ° ");
}

public static void main(String[] args) {
    new Frame0509 組合框 1();
}
}
```

```
public class Frame0509 組合框 2 extends JFrame {
    JLabel JL1;
    JComboBox JCB1;
    Container ContentPane;
    public Frame0509 組合框 2() {
        ContentPane = this.getContentPane();
    }
}
```

```

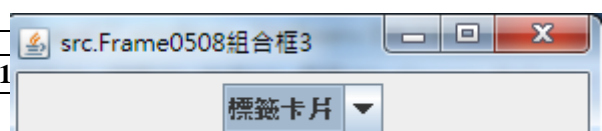
ContentPane.setLayout(null);
JL1 = new JLabel();
JCB1 = new JComboBox();
JL1.setBounds(0 · 0 · 300 · 40);
JL1.setBackground(Color.yellow);
JL1.setOpaque(true);
ContentPane.add(JL1);
JCB1.setModel(new javax.swing.DefaultComboBoxModel(new String[]{"汽車", "卡車", "戰車", "計程車", "跑車",
"迷你車"}));
JCB1.setEditable(true);
JCB1.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent evt) {
        choose(evt);
    }
});
JCB1.setBounds(0 · 60 · 150 · 40);
ContentPane.add(JCB1);
////////////////////////////////////
ContentPane.setBackground(Color.PINK);
this.setTitle(getClass().getName().toString());
this.setBounds(200 · 100 · 300 · 250);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
public void choose(ItemEvent evt) {
    int[] ar1 = {10 · 20 · 30 · 40 · 50 · 60};
    JComboBox tmp = (JComboBox) evt.getSource();
    String str = (String) tmp.getSelectedItem();
    JL1.setText("您選擇了" + str + ar1[tmp.getSelectedIndex()] + " ° ");
}
public static void main(String[] args) {
    new Frame0509 組合框 2();
}
}

```

```

public class Frame0509 組合框 3 extends JFrame {
    public JButton JB1 · JB2 · JB3;
    public JLabel JL1 · JL2;
    public JComboBox JCB1;
    public JPanel JP1 · JP2 · JP3 · cards;
    public Container ContentPane;

```



```
public Frame0509 組合框 3() {
    initComponents();
}

private void initComponents() {
    ContentPane = this.getContentPane();
    CardLayout layout = new CardLayout();
    JB1 = new JButton("按鈕一");
    JB2 = new JButton("按鈕二");
    JB3 = new JButton("按鈕三");
    JL1 = new JLabel("標籤一");
    JL2 = new JLabel("標籤二");
    JP1 = new JPanel();
    JP2 = new JPanel();
    JP3 = new JPanel();
    cards = new JPanel();
    JCB1 = new JComboBox(new String[]{"按鈕卡片", "標籤卡片"});
    JCB1.setEditable(false);
    JCB1.addItemListener(new ItemListener() {
        public void itemStateChanged(ItemEvent evt) {
            choose(evt);
        }
    });
    //組合框
    JP1.add(JCB1);
    // 第一組卡片
    JP2.add(JB1);
    JP2.add(JB2);
    JP2.add(JB3);
    // 第二組卡片
    JP3.add(JL1);
    JP3.add(JL2);
    cards.setLayout(layout);
    cards.add(JP2, "按鈕卡片");
    cards.add(JP3, "標籤卡片");
    ContentPane.add(JP1, BorderLayout.NORTH);
    ContentPane.add(cards, BorderLayout.SOUTH);
    this.setTitle(getClass().getName());
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setSize(300, 150);
    this.setVisible(true);
}

public void choose(ItemEvent evt) {
    CardLayout cl = (CardLayout) (cards.getLayout());
```

```

        cl.show(cards · (String) evt.getItem());
    }

    public static void main(String[] args) {
        new Frame0509 組合框 3();
    }
}

```

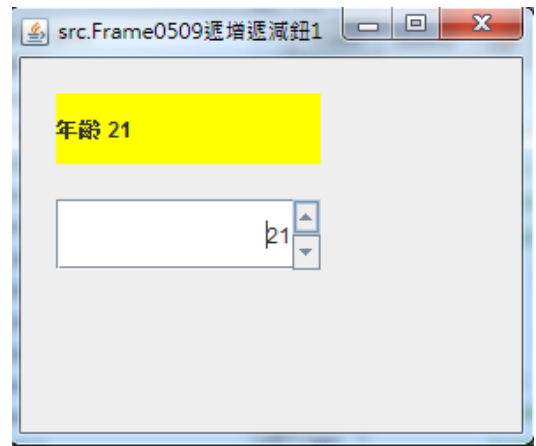
遞增遞減鈕

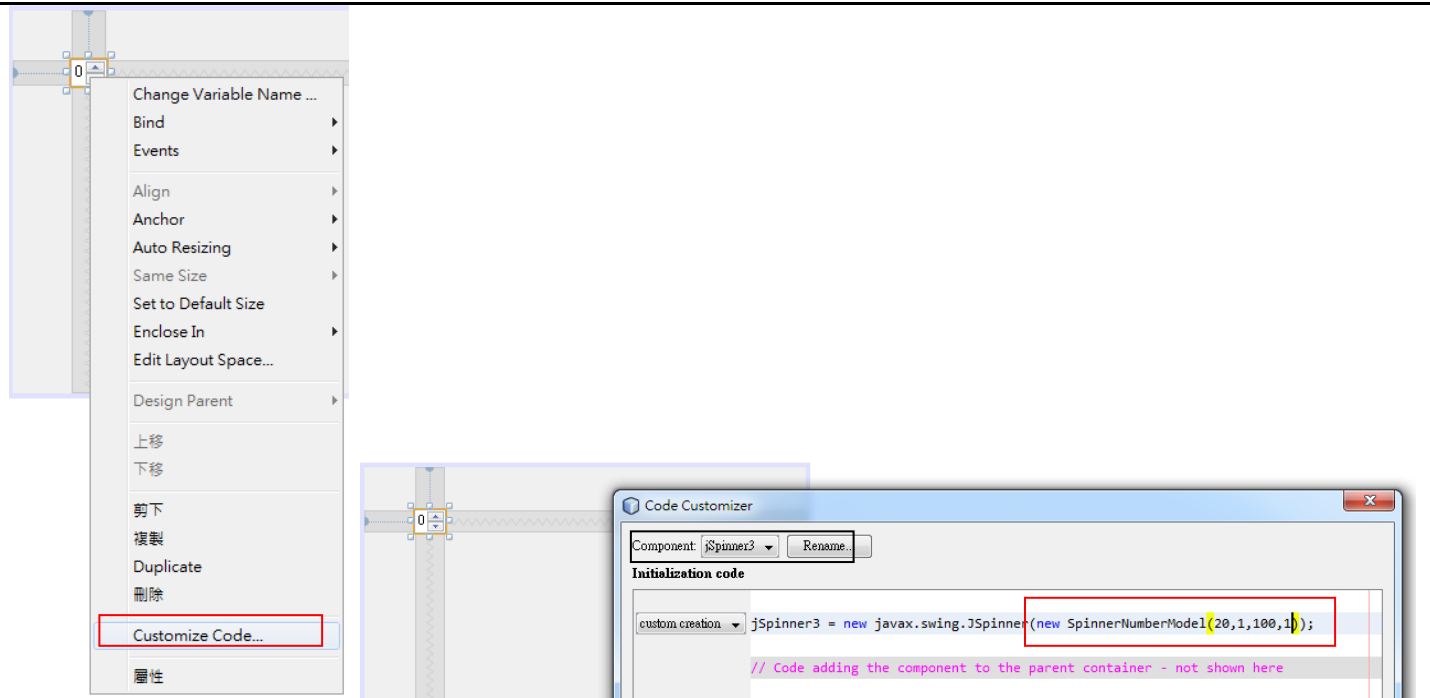
```

public class Frame0510 遞增遞減鈕 1 extends JFrame {
    public JLabel JL1;
    public JSpinner jspinner1;
    public Container ContentPane;
    public Frame0510 遞增遞減鈕 1() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        JL1 = new JLabel("年齡 ");
        jspinner1 = new JSpinner(new SpinnerNumberModel(20 · 1 · 100 · 1));
        JL1.setBackground(Color.yellow);
        JL1.setOpaque(true);
        JL1.setBounds(20 · 20 · 150 · 40);
        jspinner1.setBounds(20 · 80 · 150 · 40);

        jspinner1.addChangeListener(new ChangeListener() {
            public void stateChanged(ChangeEvent evt) {
                chose(evt);
            }
        });
        ContentPane.add(JL1);
        ContentPane.add(jspinner1);
        this.setTitle(getClass().getName());
        this.setBounds(200 · 100 · 300 · 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
    public void chose(ChangeEvent evt) {
        JL1.setText("年齡 ");
        int number = (Integer) jspinner1.getValue();
        JL1.setText(JL1.getText() + String.valueOf(number));
    }
    public static void main(String[] args) {
        new Frame0510 遞增遞減鈕 1();
    }
}

```





```

public class Frame0510 遞增遞減鈕 2 extends JFrame {
    public JLabel JL1;
    public JSpinner jspinner1;
    public Container ContentPane;
    public Frame0510 遞增遞減鈕 2() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        JL1 = new JLabel("年齡 ");
        jspinner1 = new JSpinner();
        JL1.setBackground(Color.yellow);
        JL1.setOpaque(true);
        jspinner1.setModel(new javax.swing.SpinnerNumberModel(20, 1, 100, 1));
        JL1.setBounds(20, 20, 150, 40);
        jspinner1.setBounds(20, 80, 150, 40);
        jspinner1.addChangeListener(new ChangeListener() {
            public void stateChanged(ChangeEvent evt) {
                chose(evt);
            }
        });
        ContentPane.add(JL1);
        ContentPane.add(jspinner1);
        this.setTitle(getClass().getName());
        this.setBounds(200, 100, 300, 250);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
    public void chose(ChangeEvent evt) {
        JL1.setText("年齡 ");
        int number = (Integer) jspinner1.getValue();
        JL1.setText(JL1.getText() + String.valueOf(number));
    }
}

```

```

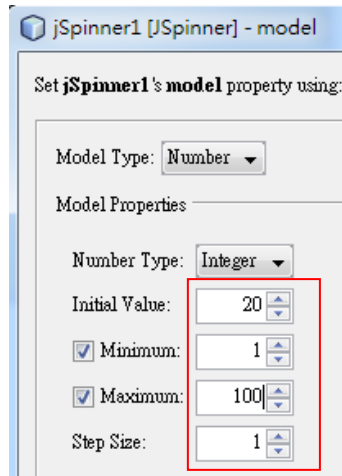
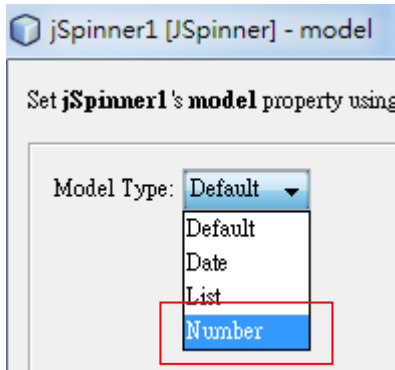
    }
    public static void main(String[] args) {
        new Frame0510 遞增遞減鈕 20;
    }
}

```



model

[SpinnerModel]



功能表

```

public class Frame0511 功能表 1 extends JFrame {
    public JLabel JL1;
    public JMenuBar menubar;
    public JMenu mn1 · mn2 · mn3 · mn4;
    public JMenuItem mi1 · mi2 · mi3 · mi4 · mi5 · mi6;
    public Container ContentPane;

    public Frame0511 功能表 1() {
        initComponents();
    }

    private void initComponents() {
        ContentPane = this.getContentPane();
        JL1 = new JLabel("歡迎光臨。");
        mn1 = new JMenu("主選項 1");
        mn2 = new JMenu("主選項 2");
        mn3 = new JMenu("子選項 1");
        mn4 = new JMenu("子選項 2");
        mi1 = new JMenuItem("汽車");
        mi2 = new JMenuItem("卡車");
        mi3 = new JMenuItem("戰車");
        mi4 = new JMenuItem("計程車");
        mi5 = new JMenuItem("跑車");
        mi6 = new JMenuItem("迷你車");
        mn1.add(mi1);
        mn1.add(mi2);
        mn2.add(mn3);
        mn2.addSeparator();
        mn2.add(mn4);
        mn3.add(mi3);
        mn3.add(mi4);
    }
}

```




```
mn4.add(mi5);
mn4.add(mi6);
menubar = new JMenuBar();
menubar.add(mn1);
menubar.add(mn2);
mi1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
mi2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
mi3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
mi4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
mi5.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
mi6.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
ContentPane.add(JL1 · BorderLayout.CENTER);
ContentPane.add(menubar · BorderLayout.NORTH);
this.setTitle(getClass().getName());
this.setBounds(200 · 100 · 300 · 250);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
public void choose(ActionEvent evt) {
    JMenuItem tmp = (JMenuItem) evt.getSource();
    String str = tmp.getText();
    JL1.setText("您選擇了" + str + "。");
}
public static void main(String[] args) {
    new Frame0511 功能表 1();
}
```

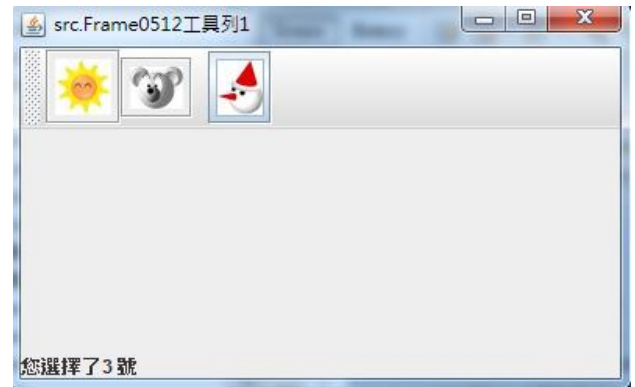
}

工具列

```

public class Frame0512 工具列 1 extends JFrame {
    public JLabel JL1;
    public JPanel JP1;
    public JButton JB1 · JB2 · JB3;
    public JToolBar toolBar;
    public ImageIcon[] ic = new ImageIcon[3];
    public Container ContentPane;
    public Frame0512 工具列 1() {
        initComponents();
        JB1.setToolTipText("1 號");
        JB2.setToolTipText("2 號");
        JB3.setToolTipText("3 號");
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ic[0] = new ImageIcon(getClass().getResource("/pic/p1.jpg"));
        ic[1] = new ImageIcon(getClass().getResource("/pic/p2.jpg"));
        ic[2] = new ImageIcon(getClass().getResource("/pic/p3.jpg"));
        JP1 = new JPanel();
        JL1 = new JLabel("歡迎光臨。");
        JB1 = new JButton(ic[0]);
        JB2 = new JButton(ic[1]);
        JB3 = new JButton(ic[2]);
        toolBar = new JToolBar();
        JB1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                choose(evt);
            }
        });
        JB2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                choose(evt);
            }
        });
        JB3.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                choose(evt);
            }
        });
        toolBar.add(JB1);
        toolBar.add(JB2);
    }
}

```



```

    toolBar.addSeparator();
    toolBar.add(JB3);
    ContentPane.add(JL1 · BorderLayout.SOUTH);
    ContentPane.add(JP1 · BorderLayout.CENTER);
    ContentPane.add(toolBar · BorderLayout.NORTH);
    this.setTitle(getClass().getName());
    this.setBounds(200 · 100 · 400 · 250);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setVisible(true);
}

public void choose(ActionEvent evt) {
    int num = 0;
    JButton tmp = (JButton) evt.getSource();
    if (tmp == JB1) {
        num = 1;
    } else if (tmp == JB2) {
        num = 2;
    } else if (tmp == JB3) {
        num = 3;
    }
    JL1.setText("您選擇了" + num + " 號");
}

public static void main(String[] args) {
    new Frame0512 工具列 1();
}
}

```

垂直水平捲軸

//捲軸條的實際最大值是 maximum 減去 visible amount

// maximum 是 300 · visible amount 是 60 · 所以實際最大值是 240 · 滾動條軌道的範圍是 0 - 300

public class Frame0513 垂直水平捲軸 1 extends JFrame {

```

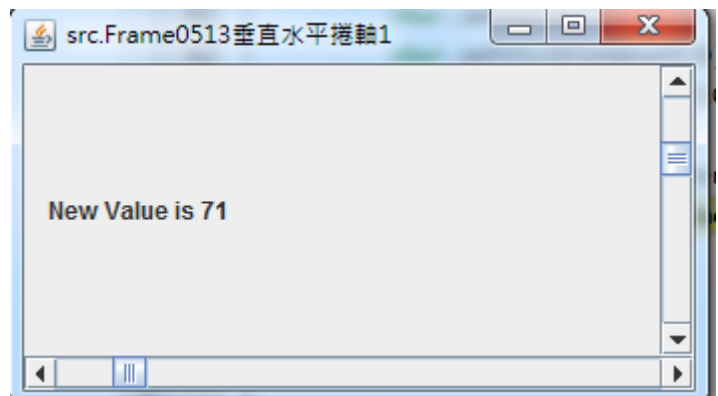
    public JLabel JL1;
    public JScrollBar hbar;
    public JScrollBar vbar;
    public Container ContentPanel;
    public Frame0513 垂直水平捲軸 1() {
        initComponents();
    }

```

```

    private void initComponents() {
        ContentPanel = this.getContentPane();
        ContentPanel.setLayout(new BorderLayout());
        JL1 = new JLabel();
        hbar = new JScrollBar();
        hbar.setOrientation(javax.swing.JScrollBar.HORIZONTAL);
    }
}

```



```

hbar.setMaximum(300);
hbar.setValue(30);
hbar.setUnitIncrement(2);
hbar.setVisibleAmount(0);
vbar = new JScrollBar();
vbar.setOrientation(javax.swing.JScrollBar.VERTICAL);
vbar.setMaximum(300);
vbar.setValue(1);
vbar.setVisibleAmount(0);
vbar.setUnitIncrement(10);
hbar.addAdjustmentListener(new java.awt.event.AdjustmentListener() {
    public void adjustmentValueChanged(java.awt.event.AdjustmentEvent evt) {
        choose(evt);
    }
});
vbar.addAdjustmentListener(new java.awt.event.AdjustmentListener() {
    public void adjustmentValueChanged(java.awt.event.AdjustmentEvent evt) {
        choose(evt);
    }
});
ContentPanel.add(hbar · BorderLayout.SOUTH);
ContentPanel.add(vbar · BorderLayout.EAST);
ContentPanel.add(JL1 · BorderLayout.CENTER);
this.setTitle(getClass().getName());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setBounds(300 · 300 · 350 · 200);
this.setVisible(true);
}
public void choose(AdjustmentEvent evt) {
    JL1.setText("    New Value is " + evt.getValue() + "    ");
    repaint();
}
public static void main(String s[]) {
    new Frame0513 垂直水平捲軸 1();
}
}

```

```

public class Frame0513 垂直水平捲軸 2 extends JFrame {
    public JScrollBar scrollBar;
    public JScrollPane scrollPane;
    public JLabel lbl 菜單圖片;
    public JTextField JT1;

```



```
public JTextArea textArea;
public Container ContentPane;
public ImageIcon img;
public String[] 菜單;
StringBuilder str = new StringBuilder("");
public Frame0513 垂直水平捲軸 2() {
    initComponents();
    菜單();
}
private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(null);
    lbl 菜單圖片 = new JLabel();
    lbl 菜單圖片.setToolTipText("");
    lbl 菜單圖片.setOpaque(true);
    JT1 = new JTextField();
    // (value extent · min · max )
    scrollBar = new JScrollBar(JScrollBar.HORIZONTAL · 1 · 0 · 1 · 20);
    //scrollBar = new JScrollBar();
    //scrollBar.setMaximum(20);
    //scrollBar.setMinimum(1);
    //scrollBar.setValue(1);
    //scrollBar.setOrientation(javax.swing.JScrollBar.HORIZONTAL);
    // scrollBar.setVisibleAmount(0);
    textArea = new JTextArea();
    textArea.setColumns(20);
    textArea.setRows(5);

    scrollPane = new JScrollPane();
    scrollPane.setViewportView(textArea);
    lbl 菜單圖片.setBounds(150 · 10 · 140 · 100);
    JT1.setBounds(150 · 150 · 90 · 28);
    scrollBar.setBounds(100 · 200 · 300 · 28);
    lbl 菜單圖片.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            選菜(evt);
        }
    });
    scrollBar.addAdjustmentListener(new AdjustmentListener() {
        public void adjustmentValueChanged(AdjustmentEvent evt) {
            菜單展示(evt);
        }
    });
};
```

```
ContentPane.add(lbl 菜單圖片);
ContentPane.add(JT1);
ContentPane.add(scrollBar);
scrollPane.setBounds(100 · 250 · 300 · 200);
ContentPane.add(scrollPane);
this.setTitle(getClass().getName());
this.setBounds(200 · 100 · 500 · 500);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
public void 菜單展示(AdjustmentEvent evt) {
    菜單();
}
public void 菜單() {
    JT1.setText("pic" + scrollBar.getValue() + ".jpg");
    img = new ImageIcon(getClass().getResource("/pic/pic" + scrollBar.getValue() + ".jpg"));
    lbl 菜單圖片.setIcon(img);
}
public void 選菜(MouseEvent evt) {
    str.append(找品名價錢("pic" + scrollBar.getValue() + ".jpg")).append("\n");
    textArea.setText(str.toString());
}
public String 找品名價錢(String strx) {
    try {
        FileReader fr = new FileReader("menu.txt");
        BufferedReader br = new BufferedReader(fr);
        String[] str1;
        String data;
        while ((data = br.readLine()) != null) {
            str1 = data.split(" · ");
            if (str1[1].equals(strx)) {
                return str1[1] + " · " + str1[0] + " · " + str1[2];
            }
        }
    } catch (IOException e) {
        System.out.println(e);
    }
    return "";
}
public static void main(String args[]) {
    new Frame0513 垂直水平捲軸 2();
}
}
```

```
public class Frame0514 頁籤 1 extends JFrame {
```

```
    public JTabbedPane tabbedPane;
```

```
    public JPanel JP1;
```

```
    public JPanel JP2;
```

```
    public JPanel JP3;
```

```
    public JButton btnCenter;
```

```
    public JButton btnEast;
```

```
    public JButton btnNorth;
```

```
    public JButton btnSouth;
```

```
    public JButton btnWest;
```

```
    public JLabel lblField1;
```

```
    public JLabel lblField2;
```

```
    public JLabel lblField3;
```

```
    public JLabel lblPassword;
```

```
    public JLabel lblUsername;
```

```
    public JScrollPane scrollPane1;
```

```
    public JScrollPane scrollPane2;
```

```
    public JScrollPane scrollPane3;
```

```
    public JTextArea txtArea1;
```

```
    public JTextArea txtArea2;
```

```
    public JTextArea txtArea3;
```

```
    public JPasswordField txtPassword;
```

```
    public JTextField txtUsername;
```

```
    public Container ContentPane;
```

```
    public Frame0514 頁籤 1() {
```

```
        initComponents();
```

```
    }
```

```
    private void initComponents() {
```

```
        ContentPane = this.getContentPane();
```

```
        ContentPane.setLayout(new BorderLayout());
```

```
        tabbedPane = new JTabbedPane();
```

```
        JP1 = new JPanel();
```

```
        JP2 = new JPanel();
```

```
        JP3 = new JPanel();
```

```
        tabbedPane.addTab("Page 1" · JP1);
```

```
        tabbedPane.addTab("Page 2" · JP2);
```

```
        tabbedPane.addTab("Page 3" · JP3);
```

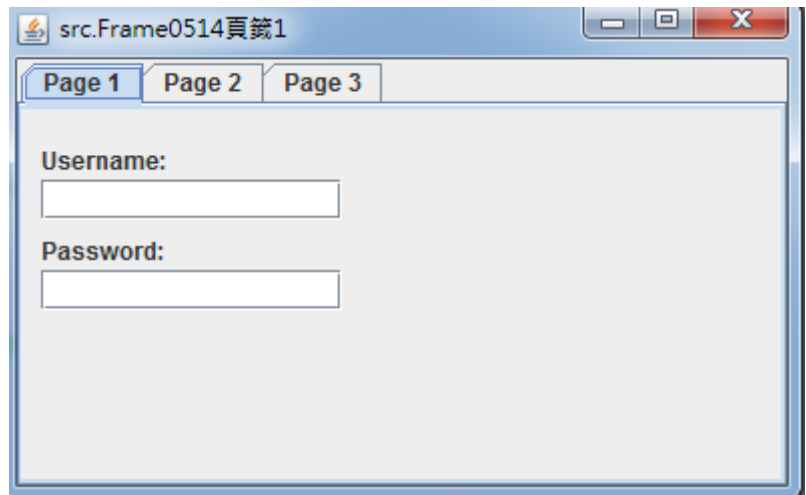
```
        lblUsername = new JLabel("Username:");
```

```
        txtUsername = new JTextField();
```

```
        lblPassword = new JLabel("Password:");
```

```
        txtPassword = new JPasswordField();
```

```
        btnCenter = new JButton("Center");
```



```
btnNorth = new JButton("North");
btnSouth = new JButton("South");
btnEast = new JButton("East");
btnWest = new JButton("West");
lblField1 = new JLabel("Field1");
txtArea1 = new JTextArea();
scrollPane1 = new JScrollPane();
scrollPane1.setViewportView(txtArea1);
lblField2 = new JLabel("Field2");
txtArea2 = new JTextArea();
scrollPane2 = new JScrollPane();
scrollPane2.setViewportView(txtArea2);
lblField3 = new JLabel("Field3");
txtArea3 = new JTextArea();
scrollPane3 = new JScrollPane();
scrollPane3.setViewportView(txtArea3);
////////////////////////////////////
lblUsername.setBounds(10 · 15 · 150 · 20);
lblPassword.setBounds(10 · 60 · 150 · 20);
txtUsername.setBounds(10 · 35 · 150 · 20);
txtPassword.setBounds(10 · 80 · 150 · 20);
JP1.setLayout(null);
JP1.add(lblUsername);
JP1.add(txtUsername);
JP1.add(lblPassword);
JP1.add(txtPassword);
////////////////////////////////////
JP2.setLayout(new BorderLayout());
JP2.add(btnNorth · BorderLayout.NORTH);
JP2.add(btnSouth · BorderLayout.SOUTH);
JP2.add(btnEast · BorderLayout.EAST);
JP2.add(btnWest · BorderLayout.WEST);
JP2.add(btnCenter · BorderLayout.CENTER);
////////////////////////////////////
JP3.setLayout(new GridLayout(3 · 2));
JP3.add(lblField1);
JP3.add(scrollPane1);
JP3.add(lblField2);
JP3.add(scrollPane2);
JP3.add(lblField3);
JP3.add(scrollPane3);
ContentPane.add(tabbedPane · BorderLayout.CENTER);
this.setTitle(getClass().getName());
```



```

this.setBounds(200 · 100 · 400 · 250);

this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

this.setVisible(true);

}

public static void main(String args[]) {

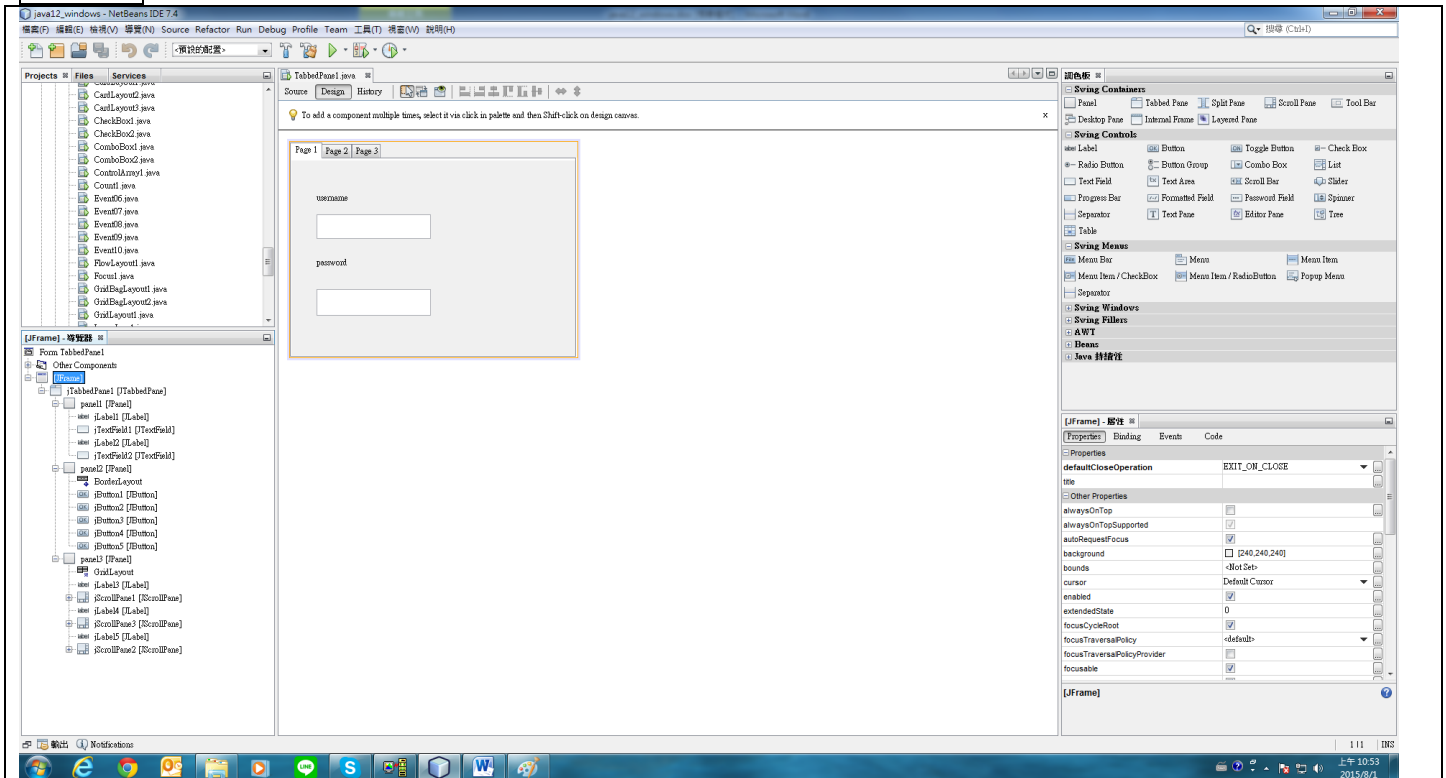
    new Frame0514 頁籤 1();

}

}

```

頁籤→視窗



顏色對話框

```

public class Frame0515 顏色對話框 1 extends JFrame {

    public JButton JB1;

    public JColorChooser jColorChooser1;

    public Color color;

    public Container ContentPane;

    public Frame0515 顏色對話框 1() {

        initComponents();

    }

    private void initComponents() {

        ContentPane = this.getContentPane();

        ContentPane.setLayout(new BorderLayout());

        JB1 = new JButton("Get Color");

        jColorChooser1 = new JColorChooser();

        JB1.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent evt) {

                choose(evt);

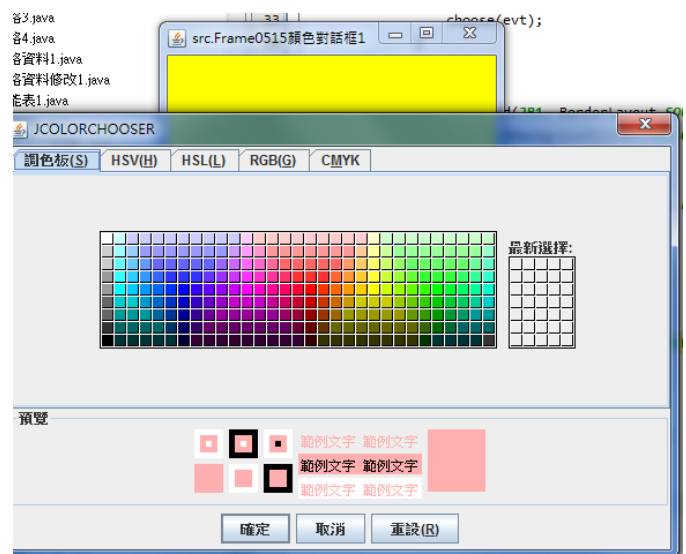
            }

        });

    }

}

```



```

});
ContentPane.add(JB1, BorderLayout.SOUTH);
ContentPane.setBackground(Color.YELLOW);
this.setTitle(getClass().getName());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setBounds(300, 300, 300, 200);
this.setVisible(true);
}

public void choose(ActionEvent evt) {
    color = jColorChooser1.showDialog(rootPane, "JCOLORCHOOSER", Color.pink);
    ContentPane.setBackground(color);
}

public static void main(String[] args) {
    new Frame0515 顏色對話框 1();
}
}

```

訊息框 · 確認框

```

public class Frame0601 訊息框 1 extends JFrame {
    public JPanel JP1, JP2;
    public JButton JB1, JB2, JB3, JB4, JB5, JB6;
    public Container ContentPane;
    public Frame0601 訊息框 1() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        JP1 = new JPanel();
        JP2 = new JPanel();
        JB1 = new JButton("question");
        JB2 = new JButton("information");
        JB3 = new JButton("warning");
        JB4 = new JButton("error");
        JB5 = new JButton("YES_NO_CANCEL");
        JB6 = new JButton("OK_CANCEL");
        JB1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                choose(evt);
            }
        });
        JB2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                choose(evt);
            }
        });
    }
}

```



```
});
JB3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
JB4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
JB5.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
JB6.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
JP1.setLayout(new GridLayout(1 · 0 · 15 · 0));
JP1.add(JB1);
JP1.add(JB2);
JP1.add(JB3);
JP1.add(JB4);
JP2.setLayout(new GridLayout(1 · 0 · 15 · 0));
JP2.add(JB5);
JP2.add(JB6);
ContentPane.add(JP1 · BorderLayout.NORTH);
ContentPane.add(JP2 · BorderLayout.SOUTH);
this.setTitle(getClass().getName());
this.setBounds(200 · 100 · 400 · 250);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

public void choose(ActionEvent evt) {
    JButton tmp = (JButton) evt.getSource();
    if (tmp == JB1) {
        JOptionPane.showMessageDialog(ContentPane · "quuestion" · "question" · JOptionPane.QUESTION_MESSAGE);
    }
    if (tmp == JB2) {
        JOptionPane.showMessageDialog(ContentPane · "information" · "information" ·
```

```

JOptionPane.INFORMATION_MESSAGE);
    }
    if (tmp == JB3) {
        JOptionPane.showMessageDialog(ContentPane, "warning", "warning", JOptionPane.WARNING_MESSAGE);
    }
    if (tmp == JB4) {
        JOptionPane.showMessageDialog(ContentPane, "error", "error", JOptionPane.ERROR_MESSAGE);
    }
    if (tmp == JB5) {
        int res = JOptionPane.showConfirmDialog(ContentPane, "YES_NO_CANCEL", "YES_NO_CANCEL",
JOptionPane.YES_NO_CANCEL_OPTION);
        if (res == JOptionPane.YES_OPTION) {
            JOptionPane.showMessageDialog(ContentPane, "yes", "yes", JOptionPane.INFORMATION_MESSAGE);
        }
    }
    if (tmp == JB6) {
        int res = JOptionPane.showConfirmDialog(ContentPane, "OK_CANCEL", "OK_CANCEL",
JOptionPane.OK_CANCEL_OPTION);
        if (res == JOptionPane.OK_OPTION) {
            JOptionPane.showMessageDialog(ContentPane, "ok", "ok", JOptionPane.INFORMATION_MESSAGE);
        }
    }
}

public static void main(String[] args) {
    new Frame0601 訊息框 1();
}
}

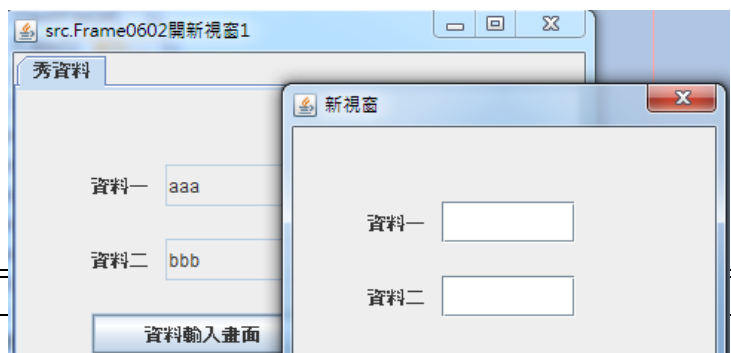
```

開新視窗

```

public class Frame0602 開新視窗 1 extends JFrame {
    public Container ContentPane;
    public JLabel JL1, JL2;
    public JTextField JT1, JT2;
    public JButton JB1;
    public JTabbedPane tabbedPane;
    public JPanel JP1;
    public Frame0602 開新視窗 1() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(new BorderLayout());
        JP1 = new JPanel();
        tabbedPane = new JTabbedPane();
        tabbedPane.addTab("秀資料", JP1);
        JL1 = new JLabel("資料一");
        JT1 = new JTextField("");
    }
}

```



```

    JL2 = new JLabel("資料二");
    JT2 = new JTextField("");
    JB1 = new JButton("資料輸入畫面");
    JT1.setEditable(false);
    JT2.setEditable(false);
    JL1.setBounds(50 · 50 · 90 · 28);
    JT1.setBounds(100 · 50 · 90 · 28);
    JL2.setBounds(50 · 100 · 90 · 28);
    JT2.setBounds(100 · 100 · 90 · 28);
    JB1.setBounds(50 · 150 · 150 · 28);
    JB1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            開新視窗(evt);
        }
    });
    JP1.setLayout(null);
    JP1.add(JL1);
    JP1.add(JT1);
    JP1.add(JL2);
    JP1.add(JT2);
    JP1.add(JB1);
    ContentPane.add(tabbedPane · BorderLayout.CENTER);
    this.setTitle(getClass().getName());
    this.setSize(400 · 400);
    this.setLocationRelativeTo(null); //設定位置在螢幕中央
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setVisible(true);
}

public void 開新視窗(java.awt.event.ActionEvent evt) {
    JTextField[] text = {JT1 · JT2};
    Frame0602 開新視窗 1w win2 = new Frame0602 開新視窗 1w(this · true);
    win2.setLocationRelativeTo(null); //設定位置在螢幕中央
    win2.setVisible(true);
    if (win2.資料取得 1() != null) {
        for (int i = 0; i < text.length; i++) {
            text[i].setText(win2.資料取得 1()[i]);
        }
    }
}

public static void main(String[] args) {
    new Frame0602 開新視窗 1();
}
}

```

```

public class Frame0602 開新視窗 1w extends javax.swing.JDialog {
    public String[] 資料 1 = null;
    public Container ContentPane;
    public JLabel JL1 · JL2;
    public JTextField JT1 · JT2;
    public JButton JB1;
    public Frame0602 開新視窗 1w(java.awt.Frame parent · boolean modal) {

```

```

    super(parent · modal);
    initComponents();
}
private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(null);
    JL1 = new JLabel("資料一");
    JT1 = new JTextField("");
    JL2 = new JLabel("資料二");
    JT2 = new JTextField("");
    JB1 = new JButton("確定");
    JL1.setBounds(50 · 50 · 90 · 28);
    JT1.setBounds(100 · 50 · 90 · 28);
    JL2.setBounds(50 · 100 · 90 · 28);
    JT2.setBounds(100 · 100 · 90 · 28);
    JB1.setBounds(50 · 200 · 90 · 28);
    JB1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            確定(evt);
        }
    });
    ContentPane.add(JL1);
    ContentPane.add(JT1);
    ContentPane.add(JL2);
    ContentPane.add(JT2);
    ContentPane.add(JB1);
    this.setTitle("新視窗");
    this.setSize(300 · 300);
    this.setVisible(false);
}
public void 確定(java.awt.event.ActionEvent evt) {
    this.資料 1 = new String[]{JT1.getText() · JT2.getText()};
    this.dispose();
}
public String[] 資料取得 1() {
    return 資料 1;
}
}

```

內部視窗

```

public class Frame0603 內部視窗 1 extends JFrame {
    public JButton JB1;
    public JDesktopPane jdp;
    public Container ContentPane;
    public static int count;
    public Frame0603 內部視窗 1() {

```



```

    initComponents();
}

private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(new BorderLayout());
    JB1 = new JButton("New Frame");
    jdp = new JDesktopPane();
    JB1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            choose(evt);
        }
    });
    ContentPane.add(JB1, BorderLayout.SOUTH);
    ContentPane.add(jdp);
    this.setTitle(getClass().getName());
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setSize(400, 300);
    this.setVisible(true);
}

public void choose(ActionEvent evt) {
    JInternalFrame jif;
    jif = new JInternalFrame("Frame " + (count++) + " true true true true");
    Container icp = jif.getContentPane();
    JButton ibtn = new JButton("JInternalFrame Button");
    icp.add(ibtn, BorderLayout.SOUTH);
    jdp.add(jif);
    jif.setSize(200, 150);
    jif.setVisible(true);
}

public static void main(String[] args) {
    new Frame0603 內部視窗 1();
}
}

```

背景圖片

```

public class Frame0604 背景設定 1 extends JFrame {
    public JPanel ContentPane;
    public JLayeredPane jlp;
    public JButton JB1;
    public JLabel background;
    public ImageIcon img;

```



```
public Frame0604 背景設定 1() {
    initComponents();
}

private void initComponents() {
    //opaque 不透明
    ContentPane = (JPanel) this.getContentPane();
    ContentPane.setOpaque(false);
    ContentPane.setLayout(null);
    jlp = this.getLayeredPane();
    background = new JLabel();
    JB1 = new JButton("按鈕");
    img = new ImageIcon(getClass().getResource("/pic/SunSet.jpg"));
    background.setIcon(img);
    background.setBounds(0 · 0 · img.getIconWidth() · img.getIconHeight());
    JB1.setBounds(20 · 20 · 90 · 28);
    //jlp.add(background · new Integer(Integer.MIN_VALUE));
    jlp.add(background);
    ContentPane.add(JB1);
    this.setTitle(getClass().getName());
    this.setBounds(200 · 100 · img.getIconWidth() · img.getIconHeight());
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setVisible(true);
}

public static void main(String[] args) {
    new Frame0604 背景設定 1();
}
}
```

```
public class Frame0604 背景設定 2 extends JFrame {
    //創建一個 JLayeredPane 用於分層的。
    public JLayeredPane jlp;
    //創建一個 Panel 和一個 Label 用於存放圖片 · 作為背景。
    public JPanel panel;
    public JLabel background;
    public ImageIcon img;
    public JButton JB1;
    public Frame0604 背景設定 2() {
        initComponents();
    }
    private void initComponents() {
        jlp = new JLayeredPane();
        panel = new JPanel();
        background = new JLabel();
        JB1 = new JButton("測試按鈕");
    }
}
```



```

img = new ImageIcon(getClass().getResource("/pic/Sunset.jpg"));
background.setIcon(img);
panel.setBounds(0 · 0 · img.getIconWidth() · img.getIconHeight());
JB1.setBounds(20 · 20 · 90 · 28);
panel.add(background);
//將 jp 放到最底層
jlp.add(panel · JLayeredPane.DEFAULT_LAYER);
//將 jb 放到高一層的地方
jlp.add(JB1 · JLayeredPane.MODAL_LAYER);
this.setLayeredPane(jlp);
this.setTitle(getClass().getName());
this.setBounds(300 · 300 · img.getIconWidth() · img.getIconHeight());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

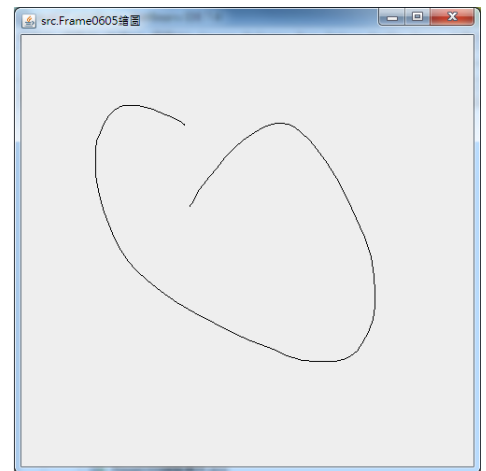
public static void main(String[] args) {
    new Frame0604 背景設定 2();
}
}

```

```

public class Frame0605 繪圖 extends JFrame {
    public int x1 · x2 · y1 · y2;
    public Container ContentPane;
    public Frame0605 繪圖() {
        initComponents();
    }
    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        ContentPane.addMouseListener(new MouseMotionAdapter() {
            public void mouseDragged(MouseEvent evt) {
                paint1(evt);
            }
        });
        ContentPane.addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent evt) {
                paint2(evt);
            }
        });
        this.setTitle(getClass().getName());
        this.setSize(500 · 500);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
}

```



```

public void paint1(MouseEvent evt) {
    Graphics g = getGraphics();
    x2 = evt.getX(); // 取得拖曳滑鼠時的 x 座標
    y2 = evt.getY(); // 取得拖曳滑鼠時的 y 座標
    g.drawLine(x1 · y1 · x2 · y2); // 繪出(x1 · y1)到(x2 · y2)的連線
    x1 = x2; // 更新繪圖起始點的 x 座標
    y1 = y2; // 更新繪圖起始點的 y 座標
}

public void paint2(MouseEvent evt) {
    x1 = evt.getX(); // 取得滑鼠按下時的 x 座標 (繪圖起始點的 x 座標)
    y1 = evt.getY(); // 取得滑鼠按下時的 y 座標 (繪圖起始點的 y 座標)
}

public static void main(String args[]) {
    new Frame0605 繪圖();
}
}

```

猜拳遊戲

```

public class Frame0606 猜拳遊戲 extends JFrame {
    public JLabel lblCompute1 · lblPerson1;
    public JLabel lblComputer2 · lblPerson2;
    public JLabel message;
    public JButton btn0 · btn1 · btn2;
    public Container ContentPane;
    public ImageIcon MyImg;
    public String person;
    public String m;
    public String b;
    public int i;
    public int j;
    public String[] arr = new String[3];
    public Frame0606 猜拳遊戲() {
        initComponents();
    }

    private void initComponents() {
        ContentPane = this.getContentPane();
        ContentPane.setLayout(null);
        lblCompute1 = new JLabel("電腦出的是 :");
        lblPerson1 = new JLabel("你出的是 :");
        lblComputer2 = new JLabel();
        lblPerson2 = new JLabel();
        btn0 = new JButton("石頭");
        btn1 = new JButton("剪刀");
        btn2 = new JButton("布");
    }
}

```



```
message = new JLabel("");
lblCompute1.setBounds(50 · 50 · 100 · 50);
lblComputer2.setBounds(150 · 50 · 100 · 50);
lblPerson1.setBounds(50 · 120 · 100 · 50);
lblPerson2.setBounds(150 · 120 · 100 · 60);
message.setBounds(150 · 200 · 100 · 50);
btn0.setBounds(280 · 50 · 60 · 40);
btn1.setBounds(280 · 105 · 60 · 40);
btn2.setBounds(280 · 160 · 60 · 40);
btn0.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
btn1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
btn2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose(evt);
    }
});
ContentPane.add(lblCompute1);
ContentPane.add(lblComputer2);
ContentPane.add(lblPerson1);
ContentPane.add(lblPerson2);
ContentPane.add(message);
ContentPane.add(btn0);
ContentPane.add(btn1);
ContentPane.add(btn2);
this.setBounds(100 · 100 · 400 · 300);
this.setTitle(getClass().getName());
this.setDefaultCloseOperation(EXIT_ON_CLOSE);
this.setVisible(true);
}

public void choose(ActionEvent evt) {
    JButton tmp = (JButton) evt.getSource();
    if (tmp == btn0) {
        j = 0;
        b = btn0.getText();
    }
}
```

```
        if (tmp == btn1) {
            j = 1;
            b = btn1.getText();
        }
        if (tmp == btn2) {
            j = 2;
            b = btn2.getText();
        }
        lblPerson2.setText(b);
        lblPerson2.setIcon(new ImageIcon(getClass().getResource("/pic/" + j + ".jpg")));
        Win(person); //取電腦亂數
        lblComputer2.setIcon(new ImageIcon(getClass().getResource("/pic/" + i + ".jpg")));
        lblComputer2.setText(arr[i]);
        System.out.println(arr[i]);
    }

    public void Win(String person) {
        arr[0] = "石頭";
        arr[1] = "剪刀";
        arr[2] = "布";
        person = arr[j];
        i = (int) (Math.random() * 3);
        String computer = arr[i];
        //判斷輸贏
        if (computer.equals(person)) {
            message.setText("平手");
        } else if (computer.equals("剪刀") && person.equals("石頭") || computer.equals("石頭") && person.equals("布") ||
computer.equals("布") && person.equals("剪刀")) {
            message.setText("你贏了");
        } else {
            message.setText("你輸了");
        }
    }
}

public static void main(String[] args) {
    new Frame0606 猜拳遊戲();
}
}
```

計算機

```
public class Frame0607 計算機 extends JFrame {
    public JTextField T1;
    public JButton B1 · B2 · B3 · B4 · B5 · B6 · B7 · B8 · B9 · B10 ·
        B11 · B12 · B13 · B14 · B15 · B16 · B17 · B18 · B19;
    public JPanel JP1;
    public Container ContentPane;
```

```
public double result = 0 · tmp = 0;
public int opId = 0; //0:無 · 1:+ · 2:- · 3:* · 4:/
public String keyin = "0"; //要在 T1 內顯示的內容 · 一開始先設為 0
public Frame0607 計算機() {
    initComponents();
    T1.setHorizontalAlignment(JTextField.RIGHT);
    T1.setEditable(false);
    T1.setBackground(Color.WHITE);
    B1.setForeground(Color.BLUE);
    B2.setForeground(Color.BLUE);
    B3.setForeground(Color.BLUE);
    B4.setForeground(Color.BLUE);
    B5.setForeground(Color.BLUE);
    B6.setForeground(Color.BLUE);
    B7.setForeground(Color.BLUE);
    B8.setForeground(Color.BLUE);
    B9.setForeground(Color.BLUE);
    B10.setForeground(Color.BLUE);
    B11.setForeground(Color.RED);
    B12.setForeground(Color.RED);
    B13.setForeground(Color.RED);
    B14.setForeground(Color.RED);
    B15.setForeground(Color.BLUE);
    B16.setForeground(Color.RED);
    B17.setForeground(Color.BLUE);
    B18.setForeground(Color.RED);
    B19.setForeground(Color.RED);
}
private void initComponents() {
    ContentPane = this.getContentPane();
    ContentPane.setLayout(null);
    GridLayout layout = new GridLayout(5 · 4 · 5 · 5);
    T1 = new JTextField(keyin);
    JP1 = new JPanel();
    B1 = new JButton("1");
    B2 = new JButton("2");
    B3 = new JButton("3");
    B4 = new JButton("4");
    B5 = new JButton("5");
    B6 = new JButton("6");
    B7 = new JButton("7");
    B8 = new JButton("8");
    B9 = new JButton("9");
```

```
B10 = new JButton("0");
B11 = new JButton("+");
B12 = new JButton("-");
B13 = new JButton("*");
B14 = new JButton("/");
B15 = new JButton(".");
B16 = new JButton("C");
B17 = new JButton("+/-");
B18 = new JButton("<--");
B19 = new JButton("=");
T1.setBounds(20 · 30 · 250 · 28);
JP1.setBounds(20 · 80 · 250 · 150);

B1.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[1]
        數字按鈕(evt);
    }
});
B2.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[2]
        數字按鈕(evt);
    }
});
B3.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[3]
        數字按鈕(evt);
    }
});
B4.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[4]
        數字按鈕(evt);
    }
});
B5.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[5]
        數字按鈕(evt);
    }
});
B6.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[6]
        數字按鈕(evt);
    }
});
```

```
B7.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[7]
        數字按鈕(evt);
    }
});
B8.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[8]
        數字按鈕(evt);
    }
});
B9.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[9]
        數字按鈕(evt);
    }
});
B10.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[0]
        數字按鈕(evt);
    }
});
B11.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[+]
        加減乘除(evt);
    }
});
B12.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[-]
        加減乘除(evt);
    }
});
B13.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[*]
        加減乘除(evt);
    }
});
B14.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[/]
        加減乘除(evt);
    }
});
B15.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent evt) { //按[.]
        數字按鈕(evt);
    }
});
```

```
    }  
});  
B16.addMouseListener(new MouseAdapter() {  
    public void mouseClicked(MouseEvent evt) { //按[C] 清除  
        清除按鈕(evt);  
    }  
});  
B17.addMouseListener(new MouseAdapter() {  
    public void mouseClicked(MouseEvent evt) { //按[+/-] · 對結果值、或正在輸入的加(減、乘、除)數都有效  
        正負號按鈕(evt);  
    }  
});  
B18.addMouseListener(new MouseAdapter() {  
    public void mouseClicked(MouseEvent evt) { //按[←]倒退  
        倒退鍵按鈕(evt);  
    }  
});  
B19.addMouseListener(new MouseAdapter() {  
    public void mouseClicked(MouseEvent evt) { //按[=]  
        等號按鈕(evt);  
    }  
});  
JP1.setLayout(layout);  
JP1.add(B1);  
JP1.add(B2);  
JP1.add(B3);  
JP1.add(B11);  
JP1.add(B4);  
JP1.add(B5);  
JP1.add(B6);  
JP1.add(B12);  
JP1.add(B7);  
JP1.add(B8);  
JP1.add(B9);  
JP1.add(B13);  
JP1.add(B10);  
JP1.add(B15);  
JP1.add(B16);  
JP1.add(B14);  
JP1.add(B17);  
JP1.add(B18);  
JP1.add(B19);  
ContentPane.add(T1);
```



```
ContentPane.add(JP1);
this.setBounds(200 · 100 · 300 · 300);
this.setTitle(getClass().getName());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
private void 數字按鈕(java.awt.event.MouseEvent evt) {
    JButton tmp = (JButton) evt.getSource();
    if (tmp == B1) {
        setKeyin("1");
    }
    if (tmp == B2) {
        setKeyin("2");
    }
    if (tmp == B3) {
        setKeyin("3");
    }
    if (tmp == B4) {
        setKeyin("4");
    }
    if (tmp == B5) {
        setKeyin("5");
    }
    if (tmp == B6) {
        setKeyin("6");
    }
    if (tmp == B7) {
        setKeyin("7");
    }
    if (tmp == B8) {
        setKeyin("8");
    }
    if (tmp == B9) {
        setKeyin("9");
    }
    if (tmp == B10) {
        setKeyin("0");
    }
    if (tmp == B15) {
        setKeyin(".");
    }
}
private void 加減乘除(java.awt.event.MouseEvent evt) {
```

```

JButton tmp = (JButton) evt.getSource();
if (tmp == B11) {
    opId = 1;
    setTitle("計算機 +");
}
if (tmp == B12) {
    opId = 2;
    setTitle("計算機 -");
}
if (tmp == B13) {
    opId = 3;
    setTitle("計算機 *");
}
if (tmp == B14) {
    opId = 4;
    setTitle("計算機 /");
}
count();
keyin = "";
}

private void 清除按鈕(java.awt.event.MouseEvent evt) {
    opId = 0;
    tmp = 0;
    result = 0;
    T1.setText(keyin = "0");
    setTitle("計算機");
}

private void 正負號按鈕(java.awt.event.MouseEvent evt) {
    if (keyin.equals("")) { //若按了[+]、[-]、[*]、[/]後尚未輸入數字就按[+/-]
        keyin = "0";
    } else {
        if (keyin.startsWith("-")) {
            keyin = keyin.substring(1); //去頭的 - 號，再按一次就把 - 號拿掉
        } else if (!keyin.equals("0")) {
            keyin = "-" + keyin; //頭加 - 號，"0" 除外
        }
    }
    T1.setText(keyin);
}

private void 倒退鍵按鈕(java.awt.event.MouseEvent evt) {
    if (keyin.equals("")) {
        return; //此情況不提供倒退功能
    }
    //按了[+][-][*][/]其中之一，但還未開始輸入數字
}
```

```
keyin = T1.getText();
if (!keyin.equals("0")) {
    if (keyin.length() > 2) {
        if (keyin.charAt(keyin.length() - 2) == 'E') //倒數第二字元是'E'，表示指數會被刪完
        {
            keyin = keyin.substring(0, keyin.length() - 2);
        } // 'E' 也要刪掉才行，故刪 2 字元
        else {
            keyin = keyin.substring(0, keyin.length() - 1);
        }
    } else if (keyin.length() == 2) //剩 2 個字
    {
        if (keyin.startsWith("-")) {
            keyin = "0"; //是個位負數，如：-5
        } else {
            keyin = keyin.substring(0, keyin.length() - 1);
        }
        //刪後面一個字
    } else //剩 1 個字
    {
        keyin = "0";
    }
    T1.setText(keyin);
}
}

private void 等號按鈕(java.awt.event.MouseEvent evt) {
    keyin = count();
    opId = 0;
}

public void setKeyin(String s) {
    if (s.equals(".")) {
        if (keyin.indexOf('.') != -1) {
            return; //已經有小數點
        }
        if (keyin.equals("")) {
            keyin = "0";
        }
    }
}

if (keyin.equals("0")) { //現在 keyin 為 0
    if (s.equals("0")) {
        return; //原來的 keyin 已是 "0" 後面不能再加 "0"
    } else if (!s.equals(".")) {
        keyin = ""; //現在輸入的若不是"."，不能加在"0"之後，把原來的 0 清掉
    }
}
```

```
    }

    }

    keyin = keyin + s; //將現在輸入的數字加在 keyin 之後
    T1.setText(keyin);
}

public String count() {
    String resultStr;
    System.out.println("opId = " + opId + " keyin = " + keyin);
    if (!keyin.equals("")) {
        tmp = Double.parseDouble(keyin);
    }
    switch (opId) {
        case 1: // +
            result = result + tmp;
            break;
        case 2: // -
            result = result - tmp;
            break;
        case 3: /*
            if (!keyin.equals("")) {
                result = result * tmp;
            }
            break;
        case 4: // /
            if (!keyin.equals("")) {
                if (tmp == 0) {
                    JOptionPane.showMessageDialog(null, "無法除零");
                } else {
                    result = result / tmp;
                }
            }
            break;
        case 0:
            result = tmp;
    }
    if (result == (long) result) {
        resultStr = String.valueOf((long) result);
    } else {
        resultStr = String.valueOf(result);
    }
    T1.setText(resultStr);
    tmp = 0;
    return resultStr;
}
```

```
}  
public static void main(String[] args) {  
    new Frame0607 計算機();  
}  
}
```

什麼是 Swing

Swing 是 JFC 中的一部分，現在已被列入 JAVA 標準類庫裡面！JFC 是 Java Foundation Classes 的縮寫，其功能是提供程序設計者設計圖形用戶界面用的，JFC 主要包括下面的 4 個部分：

1. AWT 组件：舊有的視窗组件包
2. Swing 组件：新的視窗组件包
3. Accessibility API：提供一種更先進的溝通界面。例如：語音輸入或觸摸螢幕
4. Java 2D API：提供強大的圖形處理函数支持 Drag and Drop 功能：用於在兩個相同的 java 界面，甚至是 Java 與其他應用程序界面作數據交換的操作。

什麼是 lightweight component

1. AWT 是 heavyweight component → AWT 是 heavyweight component。由於 AWT 具有 native Code 的 C 語言成分，所以若想改變 AWT 的視窗變化時，必須編寫自己的 C 語言 native Code，然後搭配 AWT 的 native Code 與 JDK 函數庫。遇到不同的平台時，又必須重新理解與編譯自己所寫的 native code，因此 AWT 不具有跨平台特性，耗時且難以理解又耗系統資源，所以就稱之為 heavyweight component，表示 AWT 組件是不容易改變的。
2. Swing 是 lightweight component → Swing 是由純 Java Code 所寫成的。因此 Swing 解決了 Java 因視窗類而無法跨平台的問題，使視窗功能也具有跨平台與延展性。而且 Swing 不需占有太多系統資源，因此我們稱 Swing 為 lightweight component。

Swing 的版面結構

1. Swing 中的所有 lightweight component 都繼承自 JComponent (抽象類別)，除了 JFrame、JDialog、JWindow 與 JApplet 這四個組件是 heavyweight component 外！JFrame 繼承原有的 AWT 中的 Frame 類別；JApplet 繼承原有的 AWT 中的 Applet 類別。
2. JFrame、JDialog、JWindow 與 JApplet 這四個組件我們統稱為最上層 (Top-Level) 組件，因為其餘的 Swing 組件都必須依附在此四組件之一上才能顯示出來。
3. 最上層的四個組件都實作了 RootPaneContainer 介面，另外還有一個類別也實作了此介面，就是 JInternalFrame，它也是一個 lightweight component，但它不是最上層的組件，也就是說 JInternalFrame 不能單獨顯示出來，必須依附在最上層組件中。
4. 介面 RootPaneContainer 定義了各種容器取得與設置的方法，這裡的容器包括 JRootPane、GlassPane、LayeredPane、和 ContentPane。JRootPane 並不是真實的容器，我們不能在 JRootPane 上加入任何的組件！要加必須加在 LayeredPane 或者是 LayeredPane 裡的 ContentPane 上。
5. 另外，組件必須加在容器中，容器本身也是一種組件，因此，我們可以簡單的理解成：先把組件放在容器中，再把容器依附在上層組件中顯示出來！
6. 四個上層組件中，只包含 JRootPane 容器，如果要加入組件，可以先取得 JRootPane 的 GlassPane、LayeredPane、和 ContentPane 中相應的一個容器，然後進行添加組件的操作！若 JRootPane 不是我們想要的容器，要用到其它的容器，必須先把其它的容器加進來。其它的容器都直接或者間接繼承於 Container，使用 JComponent 的 getRootPane 方法可獲取包含給定組件的 JRootPane，下面，用樹形結構來簡單介紹 JRootPane 的組成結構，並非繼承結構：

JRootPane

--Glass Pane

--Layered Pane

--Content Pane

--Menu Bar

<1>.JRootPane 包含有 Glass Pane 與 Layered Pane，而 Layered Pane 又包含有 Content Pane 與 Menu Bar，其中程序設計者可選擇是否使用 Menu Bar，若沒有使用 Menu Bar，則 Content Pane 就會占據整個版面。

<2>.Glass Pane，它是一個透明的面版，主要功能是它或以捕獲 JFrame 上的任何事件。GlassPane 的默認值是不可看見的，不過可以將它設為看見的 (visible)，例如若要在 GlassPane 上做繪圖的工作，就可以在 Glass Pane 上顯示出來！

<3>.Layered Pane：它是一個可以重疊組件的面版，本身就是一個容器，可以將它看成是相當多層的置物架，每一層都可以放置物品，而且上面的物品會遮住下面的物品。

<4>.Content Pane：是最重要的，且是最常到的容器。它是 LayeredPane 中的一層，一般視為最底層。通常我們會將組件放在 ContentPane 上，而不會加在 LayeredPane 中，因為若將組件加入 LayeredPane 中必須自選管理組件間層次的关系，這會加重程序維護的困難度。若將組件放入 ContentPane，對於像工具檔 (ToolBar)、PopupMenu 等組件，Java 會自動高速其層次關係。

JFrame

1. 當建立一個 JFrame 組件時，系統會為此 JFrame 建立 JRootPane 組件，我們就可以任意地取得 JRootPane 上的 Glass Pane、Layered Pane 或 Content Pane，然後進行其它的操作。

- ```
JFrame frame = new JFrame();
Container contentPane = frame.getContentPane();
JButton button = new JButton();
contentPane.add(button);
```

- ```

graph BT
    JApplet --> Applet
    Applet --> Panel
    Panel --> Container
    JPanel --> JComponent
    JComponent --> Container
    Container --> Component
    Component --> Object
    Dialog --> Window
    JFrame --> Frame
    Frame --> Window
    Window --> Container

```
- Object
- Component 组件
- Container 容器
- Panel
- JComponent
- Window 抽象窗体
- Applet
- JPanel
- Dialog
- Frame
- JApplet
- JFrame

- 2019/4/24