

窗工具組(AWT)－

①Frame 類別

說明: ❶ 建構:Frame();

Frame(String 視窗標題);

❷方法:

| | |
|-------------------------------|--------------------------------|
| void setSize(int 寬度,int 高度) | 設定視窗大小 |
| void setLocation(int X,int Y) | 設定視窗左上角座標 |
| void setVisible(boolean) | true:顯示,false:隱藏 |
| void show() | 顯示視窗 |
| void setTitle(String 視窗標題) | 設定視窗標題 |
| void getTitle() | 取得視窗標題 |
| void setMenuBar(MenuBar 物件) | 設定視窗的功能表 |
| boolean isResizable() | 檢查是否可變視窗大小 |
| void setResizable(boolean) | true:可改變視窗大小 false:不可改變視窗大小 |
| void setBackground(Color 物件) | 設定視窗的背景色 |
| Color getBackground() | 取得視窗的背景色 |
| void setLayout(版面配置) | 設定視窗版面配置 |
| void setBounds(X,Y,寬度,高度) | 設定 Frame 位置與大小 |

②Color 類別

說明: 設定顏色方法有-

❶Color.black,Color.red,Color.yellow,...

❷Color(紅色值,綠色值,藍色值);

註:顏色的值域為 0~255

③版面配置

1.BorderLayout

說明: ❶建構:BorderLayout();

BorderLayout (int 水平間隔,int 垂直間隔);

註:設定各元件的水平與垂直間隔

❷方法:

| | |
|--------------------|---------|
| void setHgap(水平間隔) | 設定水平間隔 |
| void setVgap(垂直間隔) | 設定垂直間隔 |
| CENTER | 將元件放在中間 |

| | |
|--------------|---------|
| EAST | 將元件放在右邊 |
| NORTH | 將元件放在上方 |
| SOUTH | 將元件放在下方 |
| WEST | 將元件放在左邊 |

2.CardLayout

說明: ❶建構:CardLayout();

CardLayout (int 水平間隔,int 垂直間隔);

註:設定各元件的水平與垂直間隔

❷方法:

| | |
|---------------------------------------|---------|
| void first(Container) | 顯示第一個元件 |
| void last(Container) | 顯示最末個元件 |
| void next(Container) | 顯示下一個元件 |
| void previous(Container) | 顯示前一個元件 |
| void setHgap(水平間隔) | 設定水平間隔 |
| void setVgap(垂直間隔) | 設定垂直間隔 |
| int getHgap() | 取得水平間隔 |
| int getVgap() | 取得垂直間隔 |
| void removeLayoutComponent(元件) | 移除指定的元件 |
| void show(Container,元件) | 顯示元件 |

3.FlowLayout

說明: ❶建構:FlowLayout();

FlowLayout (int 對齊方式);

FlowLayout (int 對齊方式,int 水平間隔,int 垂直間隔);

註:對齊方式為「FlowLayout.LEFT、FlowLayout.CENTER、FlowLayout.RIGHT」

❷方法:

| | |
|--------------------------------|--------|
| int getAlignment() | 取得對齊方式 |
| void setAlignment(對齊方式) | 設定對齊方式 |
| void setHgap(水平間隔) | 設定水平間隔 |
| void setVgap(垂直間隔) | 設定垂直間隔 |
| int getHgap() | 取得水平間隔 |
| int getVgap() | 取得垂直間隔 |

4.GridLayout

說明: ❶建構:GridLayout();

GridLayout (int 列數,int 行數);

GridLayout (int 列數,int 行數,int 對齊方式,int 水平間隔,
int 垂直間隔);

註:設定各方格水平與垂直間隔

❷方法:

| | |
|----------------------------|--------|
| int getColumnns() | 取得行數 |
| int getRows() | 取得列數 |
| int getHgap() | 取得水平間隔 |
| int getVgap() | 取得垂直間隔 |
| void setColumns(行數) | 設定行數 |
| void setRows(列數) | 設定列數 |
| void setHgap(水平間隔) | 設定水平間隔 |
| void setVgap(垂直間隔) | 設定垂直間隔 |

④Panel 類別

說明: ❶Panel 可以擁有自己的屬性,包括顏色、大小、版面配置;

❷建構:Panel();

Panel(版面配置);

⑤Label 類別

說明: ❶文字標籤是顯示單列文字,但是不會產生任何事件

❷建構:Label();

Label(String 文字);

❸方法:

| | |
|-------------------------------------|----------------|
| String getText() | 取得 Label 的文字 |
| void setText(String 字串) | 設定 Label 的文字 |
| void setAlignment(對齊) | 設定 Label 的文字對齊 |
| void setFont(Font 物件) | 設定 Label 的字型樣式 |
| void setForeground(Color 物件) | 設定 Label 的文字顏色 |
| void setBackground(Color 物件) | 設定 Label 的背景色 |

註:對齊方式有「Label.CENTER,Label.LEFT,Label.RIGHT」

⑥Font 類別

說明: 設定字型樣式-

Font(字型名稱,樣式,大小);

註: ❶ 樣式有 Font.PLAIN(標準)、Font.BOLD(粗體)、
Font.ITALIC(斜體)

❷ 併用時以「|」連接 如 Font.BOLD | Font.ITALIC

⑦Button 類別

說明: ❶ 建構: Button();

Button(String 文字);

❷ 方法:

| | |
|--------------------------|---------------|
| void setLabel(字串) | 設定 Button 的標題 |
| String getLabel() | 取得 Button 的標題 |

⑧TextField 類別

說明: ❶ 建構: TextField();

TextField(int 長度);

TextField(String 文字);

TextField(String 文字,int 長度);

❷ 方法:

| | |
|-------------------------------------|---------|
| void setBackground(Color 物件) | 設定背景色 |
| String getText() | 取得文字內容 |
| void setText(字串) | 設定文字內容 |
| void selectAll() | 選取所有的文字 |
| void setEchoChar(字元) | 設定回應的字元 |
| void setEditable(boolean) | 設定是否可編輯 |

⑨TextArea 類別

說明: ❶ 建構: TextArea();

TextArea (int 列數,int 欄數); 設定欄列數

TextArea(String 文字,int 列數,int 欄數);

TextArea(String 文字,int 列數,int 欄數,int 捲軸);

❷ 方法:

| | |
|-------------------------|------|
| void append(字串) | 加入字串 |
| int getColumns() | 取得欄 |

| | |
|-----------------------------------|------------|
| int getRows() | 取得列 |
| void setText(字串) | 設定文字內容 |
| SCROLLBARS_BOTH | 設定有垂直與水平捲軸 |
| SCROLLBARS_HORIZONTAL_ONLY | 設定有水平捲軸 |
| SCROLLBARS_VERTICAL_ONLY | 設定有垂直捲軸 |
| SCROLLBARS_NONE | 設定無捲軸 |

⑩CheckBox 類別

說明: ❶建構:Checkbox();

Checkbox(String 文字);

Checkbox(Strig 文字,boolean); true:選取,false:未選

❷方法:

| | |
|---|----------|
| CheckboxGroup() | 建構群組物件 |
| void setCheckboxGroup(CheckboxGroup) | 設定群組 |
| String getLabel() | 取得標籤名稱 |
| boolean getState() | 檢查是否被選取 |
| void setState(boolean) | 設定選取或不選取 |

⑪List 類別

說明: ❶建構:List();

List(int 列數);

List((int 列數,boolean 複選);

❷方法:

| | |
|--------------------------------------|--------------|
| void add(String 選項) | 加入選項 |
| void add(String 選項,int 位置) | 加入選項到指定的位置 |
| String getItem(int 位置) | 取得指定位置的選項 |
| String[] getItems() | 取得所有選項 |
| int getItemCount() | 取得選項數目 |
| int getRows() | 取得列數 |
| int getSelectedIndex() | 取得被選取項目的索引值 |
| String getSelectedItem() | 取得被選取項目的內容 |
| String[] getSelectedItems() | 取得所有被選取項目的內容 |
| boolean isMultipleMode() | 檢查是否為複選 |
| void setMultipleMode(boolean) | 設定是否可複選 |
| void select(int) | 設定選項為被選取 |

②Choice 類別

說明: ❶建構:Choice();

❷方法:

| | |
|---------------------------------|-------------|
| void add(String) | 加入選項 |
| String getItem(int) | 取得指定位置的選項 |
| int getItemCount() | 取得選項數目 |
| int getSelectedIndex() | 取得被選取項目的索引值 |
| String getSelectedItem() | 取得被選取項目的內容 |
| void insert(String, int) | 將選項插入指定位置 |
| void select(int) | 設定選項為被選取 |
| void select(String) | 設定選項為被選取 |

③Scrollbar 類別

說明: ❶建構:Scrollbar(); 建立垂直捲軸

Scrollbar(int 方向);

Scrollbar(int 方向,int 初始值,int 可視大小,int 捲軸最小值,
int 捲軸最大值);

註: 以 HORIZONTAL, VERTICAL 設定捲軸方向

❷方法:

| | |
|---|--------------------------|
| int getMaximum() | 取得捲軸的最大值 |
| int getMinimum() | 取得捲軸的最小值 |
| int getOrientation() | 取得捲軸的方向 |
| int getValue() | 取得捲軸的數值 |
| int getVisibleAmount() | 取得捲軸的可視大小 |
| void setMaximum(int 最大值) | 設定捲軸的最大值 |
| void setMinimum(int 最小值) | 設定捲軸的最小值 |
| void setOrientation(int 方向) | 設定捲軸的方向 |
| void setValue(數值) | 設定捲軸的數值 |
| void setValues(int 數值,int 捲軸可 視大小,int 最小值,int 最大值) | 設定捲軸的數值、可視大 小、最小值、最大值 |
| int setVisibleAmount() | 設定捲軸的可視大小 |
| void setBlockIncrement(int 數值) | 設定捲軸區塊的增減量 |
| void setUnitIncrement(int 數值) | 設定捲軸的增減量 |

事件處理一

①事件類別

說明: ❶ActionEvent:

滑鼠按一下: 「Button」、「Menu」

滑鼠按兩下: 「List」的選項

按 Enter 鍵: 「TextField」中

❷AdjustmentEvent:

移動「Scrollbar」的指標

❸ItemEvent:

選擇「Choice」、「List」、「Checkbox」

❹TextEven:

改變「TextField」、「TextArea」的內容

❸TextSelectionEvent:

選取「TextField」、「TextArea」的內容

❹WindowEvent:

操作視窗

❺InputEvent:

參考 Mouse/ KeyEvent

❻MouseEvent:

滑鼠的 click、move、drag

❼KeyEvent:

按鍵的 press、release

②事件類別與監聽介面

| 介面名稱 | 抽象方法 |
|--------------------|---|
| ActionListener | actionPerformed(ActionEvent e) |
| | <div> <div>ActionEvent 類別</div> <div> int ALT_MASK:按下 Alt 鍵(1 表是,0 表否) int CTRL_MASK:按下 Ctrl 鍵 int SHIFT_MASK:按下 Shift 鍵 String getActionCommand():取得產生事件的元件標題 Object getSource():取得產生事件的元件名稱 </div> </div> |
| AdjustmentListener | adjustmentValueChanged(AdjustmentEvent e) |

| | |
|----------------------|---|
| | AdjustmentEvent 類別 int BLOCK_DECREMENT:按捲軸區塊減量位置 int BLOCK_INCREMENT:按捲軸區塊增量位置 int UNIT_DECREMENT:按捲軸單位減量位置 int UNIT_INCREMENT:按捲軸單位增量位置 int TRACK:拖曳捲軸指標 int getAdjustmentType():取得捲軸的常數 int getValue():取得捲軸所代表的數值 |
| FocusListener | focusGained(FocusEvent e) focusLost(FocusEvent e) |
| | FocusEvent 類別 int FOCUS_GAINED:取得鍵盤控制權 int FOCUS_LOST:失去鍵盤控制權 boolean isTemporary():判斷是否為暫時性的失去鍵盤控制權 |
| ItemListener | itemStateChanged(ItemEvent e) |
| | ItemEvent 類別 int SELECTED:選項被選取 int DESELECTED:選項被取消選取 Object getItem():取得產生事件的選項 ItemSelectable getItemSelectable():取得產生事件的元件 int getStateChange():取得選項的最新狀態 |
| KeyListener | keyPressed(KeyEvent e):按下按鍵 keyReleased(KeyEvent e):放開按鍵 keyTyped(KeyEvent e):輸入字元 |
| | KeyEvent 類別 char getKeyChar():取得按下的字元 char getKeyCode():取得按下的字元碼 boolean isActionKey():判斷是否按下 Action Key 註:Action Key 是指功能鍵、方向鍵... |
| MouseListener | mouseClicked(MouseEvent e):按滑鼠鍵一下 mousePressed(MouseEvent e):按下滑鼠鍵 mouseReleased(MouseEvent e):放開滑鼠鍵 mouseEntered(MouseEvent e):滑鼠移入 |

| | |
|----------------------------|---|
| | <p>mouseExited(MouseEvent e):滑鼠移出</p> <p>MouseEvent 類別</p> <p>int MOUSE_CLICKED:按滑鼠鍵一下</p> <p>int MOUSE_DRAGGED:拖曳滑鼠</p> <p>int MOUSE_ENTERED:移入元件</p> <p>int MOUSE_EXITED:移出元件</p> <p>int MOUSE_MOVED:移動滑鼠</p> <p>int MOUSE_PRESSED:按下滑鼠鍵</p> <p>int MOUSE_RELEASED:放開滑鼠鍵</p> <p>int getClickCount():取得按滑鼠的次數</p> <p>boolean isPopupTrigger():判斷是否顯示 Popup 功能表</p> <p>int getX():取得滑鼠的 X 座標</p> <p>int getY():取得滑鼠的 Y 座標</p> |
| MouseMotionListener | <p>mouseDragged(MouseEvent e):滑鼠拖曳</p> <p>mouseMoved(MouseEvent e):滑鼠移動</p> |
| TextListener | <p>textValueChanged(TextEvent e)</p> <p>TextEvent 類別</p> <p>int TEXT_VALUE_CHANGED:確定文字方塊的內容被改變</p> |
| WindowListener | <p>windowActivated(WindowEvent e)</p> <p>windowDeactivated(WindowEvent e)</p> <p>windowOpened(WindowEvent e)</p> <p>windowClosed(WindowEvent e)</p> <p>windowClosing(WindowEvent e)</p> <p>windowIconified(WindowEvent e)</p> <p>windowDeiconified(WindowEvent e)</p> <p>WindowEvent 類別</p> <p>int WINDOW_ICONIFIED:視窗最小化</p> <p>int WINDOW_DEICONIFIED:視窗還原</p> <p>int WINDOW_OPENED:視窗已開啟</p> <p>int WINDOW_CLOSING:選擇視窗關閉鈕</p> <p>int WINDOW_CLOSED:視窗已關閉</p> <p>int WINDOW_ACTIVATED:視窗取得控制權</p> <p>int WINDOW_DEACTIVATED:視窗失去控制權</p> |