以GUI為例了解物件以及Event

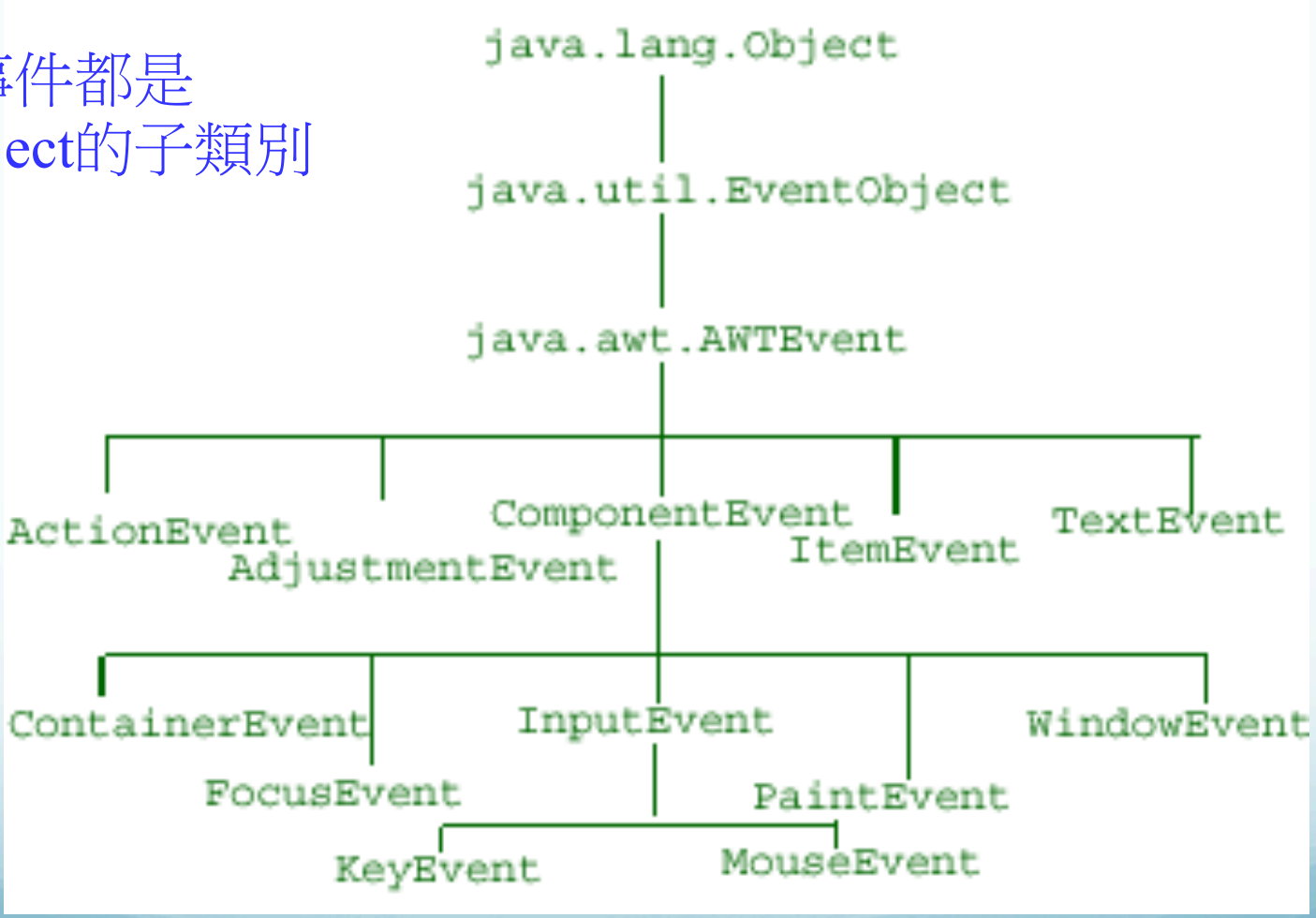
Lecturer：賀耀華

Event

- 當我們在寫程式時，多半會需要與使用者互動或回應其指令
- Java 的awt 則採用event-drivenprogramming 的方法來達成此目的，當某個特定的事件發生，就會驅動程式去執行某些特定的動作，而可與使用者產生即時的互動
- 三個要素
  - Event – 意指某個特定的事件、動作，也就是「發生了什麼事件」。例如：視窗關閉、滑鼠移動。
  - Event Source – 產生、觸發事件的元件。例如：Button
  - Event Handler – 負責接收Event object 並作處理的Method
- EventSource，產生了某個Event object，而由Event Listener負責處理這個Event

Events 以物件來表示

所有的事件都是
EventObject的子類別



- 所有的訊息都包含在java.awt.event類別庫內

以GUI為例了解物件

HelloWindow.java

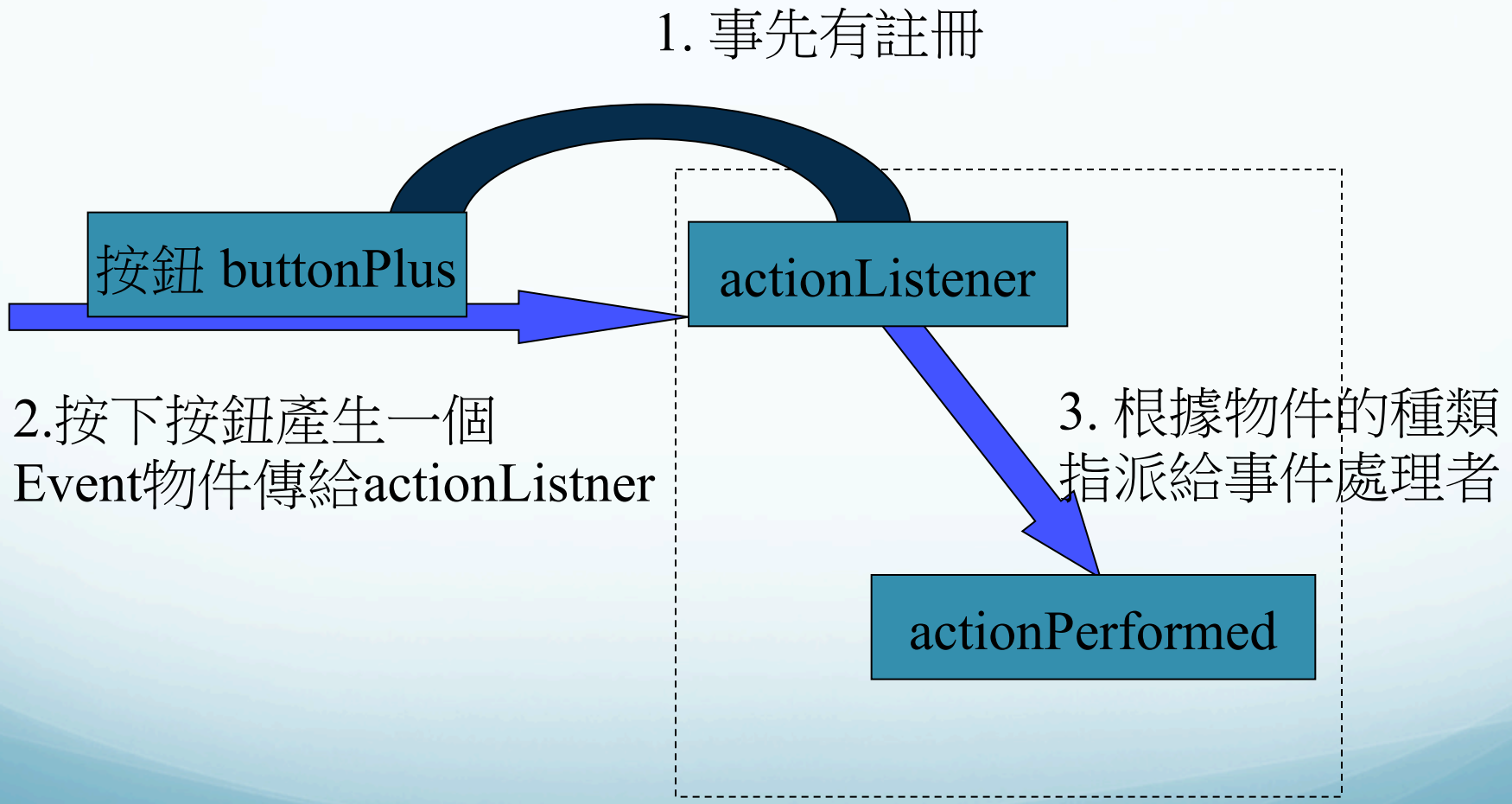

```
JFrame frame = new JFrame("HelloWindow");          //建立視窗框架
Container cp = frame.getContentPane();             //獲得內容面板
cp.setLayout(new FlowLayout(FlowLayout.LEFT));     //設定版面配置
JTextField textA = new JTextField(20);             //宣告文字元件
JButton button1 = new JButton("hi");                        //宣告按鈕元件
cp.add(textA);                                              //將元件加入面板
cp.add(button1);                                            //將元件加入面板
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);//設定關閉動作
frame.pack();                                               //調整視窗大小
frame.setVisible(true);                                     //顯示視窗
```

以GUI範例了解物件以及Event

- 以MyGUI了解Event (MyGUI.java) 作加法

```
public MyGUI()        //MyGUI.java
{
        buttonPlus.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int varA = Integer.parseInt(textA.getText());
                int varB = Integer.parseInt(textB.getText());
                Integer varC = new Integer(varA+varB);
                textC.setText(varC.toString());
            }
        });
}
```

委派事件模型


1. 事先有註冊
按鈕 buttonPlus
actionListener
2.按下按鈕產生一個
Event物件傳給actionListner
3. 根據物件的種類
指派給事件處理者
actionPerformed

系統實際運作狀況

- 當事件發生時，會有一個事件ID產生
- GUI元件使用這個ID碼，呼叫對應的事件方法
  - 假如收到有ActionEvent這種物件規格
  - 從全部已註冊的ActionListeners中，選出欲呼叫的actionPerformed() 方法

另一個版本

```
class MyListener implements ActionListener {   //介面
    public void actionPerformed(ActionEvent e) {
                    //實現這個介面一定要實作actionPerformed
        int varA = Integer.parseInt(textA.getText());
        int varB = Integer.parseInt(textB.getText());
        Integer varC = new Integer(varA+varB);
        textC.setText(varC.toString());
    }
}
public MyGUI2()   {                                  // MyGUI2.java
   MyListener listener =  new MyListener();       //
   buttonPlus.addActionListener(listener);
}
```

Event的註冊

- Event 產生時，只會通知有註冊過的Listener。所以對必須要先把Event『註冊』給要負責處理的Listner
  - 註冊所有想要擷取的事件，而當使用者啟動的事件並不是我們所想要的事件時，就不加以理會
- 程式上以XX.addXXListener 來完成註冊
  - button.addActionListener(new ActionListener() …
  - 一個event source 可以被好幾個listener 所註冊，同樣地，一個listener 也可以註冊好幾個event source
- 所有的Event Listener 都是一種interface，裡面只有定義這個Listener所提供的抽象method
  - 必須去實作出此listener interface 內所有的method

事件物件說明

事件名稱	發生事件的原因
ActionEvent	按下按鈕、或是在輸入文字方塊/選擇清單方塊時按下Enter
AdjustmentEvent	移動捲軸物件時
ItemEvent	選取核取方塊、選項鈕、下拉式清單和清單方塊
TextEvent	輸入的文字內容改變
ComponentEvent	隱藏、移動、顯示和調整元件時
ContainerEvent	新增或刪除元件
FocusEvent	元件取得或失去焦點時
KeyEvent	鍵盤按下、放開和輸入字元
MouseEvent	與滑鼠有關的行為
WindowEvent	視窗的操作，包括開、關、調整大小
PaintEvent	與繪圖有關的動作
InputEvent	它是KeyEvent和MouseEvent的父抽象類別

ActionListener

都是EventListener的子類別

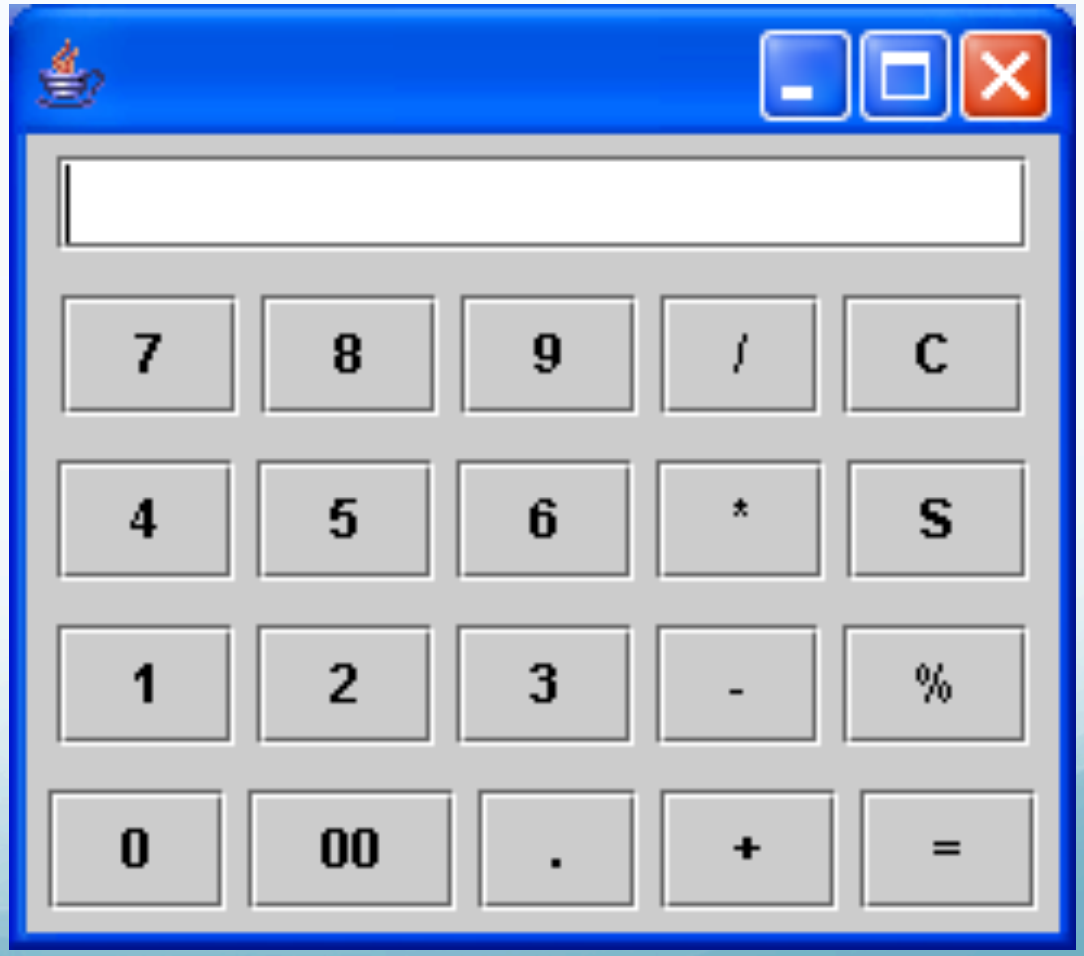
Action Type	ActionListener
ComponentEvent	ComponentListener
FocusEvent	FocusListener
KeyEvent	KeyListener
ContainerEvent	ContainerListener
WindowEvent	WindowListener
ItemEvent	ItemListener
AdjustEvent	AdjustListener
TextEvent	TextListener
ActionEvent	ActionListener

處理的方法


Button
CheckBox
Component
ActionListener
actionPerformed(ActionEvent e)
AdjustmentListener
adjustmentValueChanged(AdjustmentEvent e)
ItemListener
itemStateChanged(ItemEvent e)
KeyListener
keyTyped(KeyEvent e)
keyPressed(KeyEvent e)
keyReleased(KeyEvent e)
MouseListener
mouseClicked(MouseEvent e)
mouseEntered(MouseEvent e)
mouseExited(MouseEvent e)
mousePressed(MouseEvent e)
mouseReleased(MouseEvent e)
MouseMotionListener
mouseDragged(MouseEvent e)
mouseMoved(MouseEvent e)

Your Turn

- 實作出MyCalc小算盤 (利用MyCalc.java半成品改良)
- Form已經建好了
- 完成計算機功能
- [00]代表 00
- [/] 除兩個數字
- S取平方根
- % 例如
  - 50*10%
  - 5



問卷填寫

- Grades:
  - A: Missing 0 ~ 1 Class
  - B: Missing 1 ~ 2 Class
  - C: Missing 4 ~ more Class

- 問卷填寫
  - 最後一堂課當日開放線上填寫（填寫日期為課後十天）
  - 課後第二天會寄e-mail給上課學員
  - 原則上要上課超過50%以上才能填寫

Thank you all!