

Swing (二)

人工智慧與無線感應設備開發專班

湜憶電腦知訊顧問股份有限公司

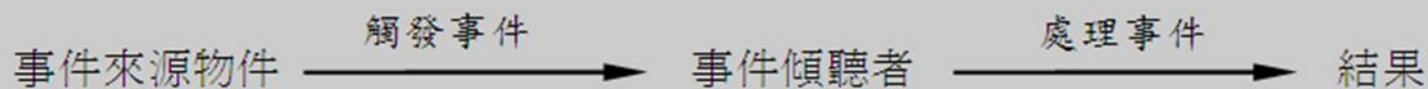
馬傳義

前言

- 圖形使用者界面的程式執行，是採用事件驅動（Event Driver）的方式來運作。
- 當程式設計師所撰寫的視窗程式執行後，程式便等開始等待使用者來觸發事件，再依據不同的事件，由程式執行相對應的方法做出回應。
- 「事件」是指，當使用者運用滑鼠、鍵盤、觸控螢幕...等周邊設備，執行點按視窗、視窗內的按鈕、文字欄位、標籤或選取功能表...，這時便會觸發到相對應的事件。
- Java處理事件驅動程式的機制是由「委派事件模式」來實作。

前言

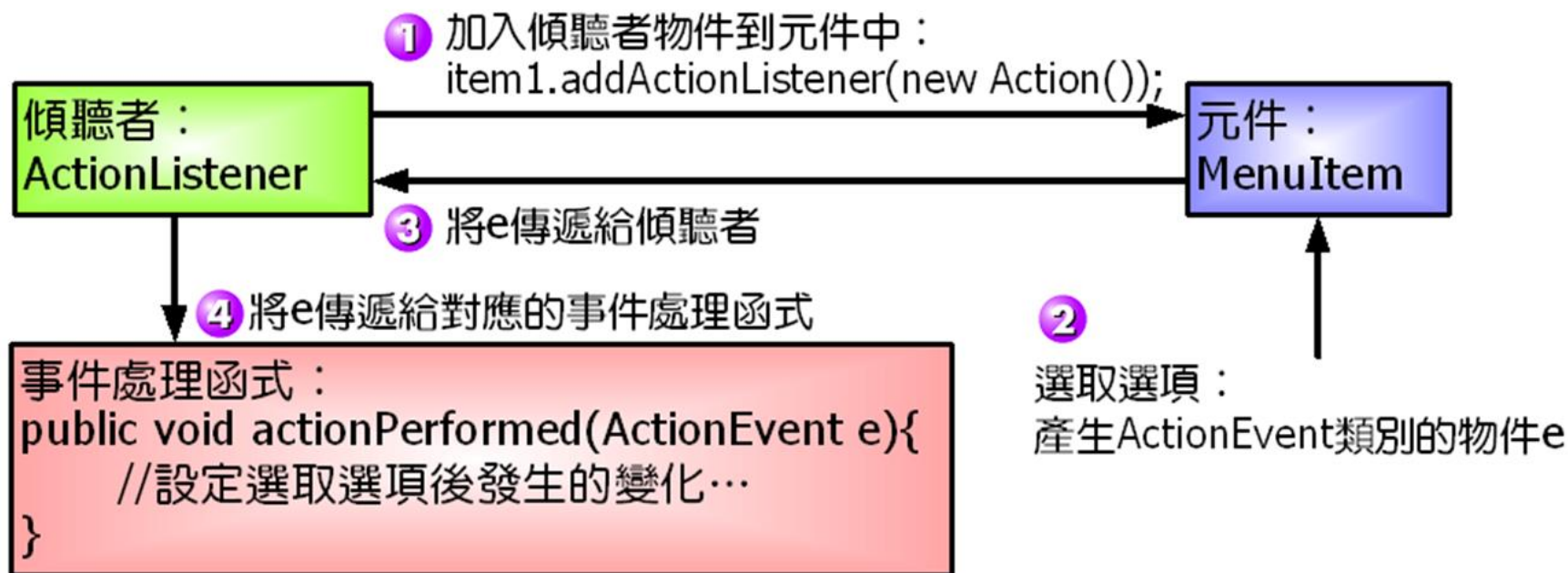
- 當使用者使用物件觸發了事件，這個「事件來源物件」轉知「事件傾聽者」接收，安排「事件處理方法」來處理事件。如下圖所示：



事件傾聽者

- 每一個視窗中的物件（包含視窗本身）都可以針對特定的事件來指定一個或多個「事件傾聽者（Listener）」物件，讓這些「事件傾聽者」物件來負責處理事件。
- 視窗中的物件可以觸發事件，因此，必須將物件個別與事件傾聽者連結在一起（即註冊事件傾聽者）。
- 傾聽者的任務是當元件發生事件時，就會產生對應事件類別的物件，然後將該物件傳遞給傾聽者，傾聽者會將發生的事件物件傳送給事件處理的方法，再由該方法做這個事件對應的動作。

事件傾聽者



以MenuItem為例

事件傾聽者

- 每一種事件類別都會對應一個傾聽者介面。
- 「事件傾聽者（ActionListener）」「介面」被定義在java.awt.event套件中，因此，
 1. 需在程式開頭處用import來匯入該套件及其中所有類別。

import java.awt.event.*;

2. 定義一個繼承JFrame類別且實作事件傾聽者介面的子類別。
3. 對事件來源物件註冊事件傾聽者（即加入傾聽者物件到元件中）。
4. 覆載要處理的事件方法。

事件傾聽者

- ◆ 步驟2 ~ 步驟4 (以「視窗」為傾聽對象) :

```
class 子類別名稱 extends JFrame implements ActionListener
{
    ⋮
    建構子()
    {
        ⋮
        事件來源的物件.addActionListener(this);
        ⋮
    }
    ⋮
    public void actionPerformed(ActionEvent e)
    {
        ⋮
    }
    ⋮
}
```

事件傾聽者

- ◆ 步驟2 ~ 步驟4 (以「元件」為傾聽對象) :

```
class 子類別名稱 extends JFrame
{
    ⋮
    建構子()
    {
        ⋮
        事件來源的物件.addActionListener(傾聽者物件名稱);
        ⋮
    }
    public ActionListener 傾聽者物件名稱=new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            ⋮
        }
    };
}
```


事件傾聽者

- ◆ 步驟2 ~ 步驟4 (匿名的「事件傾聽者」) :

```
class 子類別名稱 extends JFrame
{
    ⋮
    建構子()
    {
        事件來源物件.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                ⋮
            }
        });
    }
}
```

事件傾聽者

◆ 程式（以「視窗」為傾聽對象）：

```
package CH09_07;
```

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

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

```
class CLoginF extends JFrame implements ActionListener  
{
```

```
    private JTextField txtID, txtShow;
```

```
    private JPasswordField pwfPWS;
```

```
    private JButton btnOK, btnRST;
```

```
    CLoginF()
```

```
{
```

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

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

```
    frame.setBounds(100, 100, 235, 230);
```

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

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

事件傾聽者

```
JLabel lblTitle = new JLabel("請輸入帳號及密碼");  
lblTitle.setBounds(60, 20, 200, 20);  
frame.add(lblTitle);
```

```
JLabel lblID = new JLabel("帳號 : ");  
lblID.setBounds(30, 50, 50, 20);  
frame.add(lblID);
```

```
txtID = new JTextField();  
txtID.setBounds(70, 50, 120, 20);  
frame.add(txtID);
```

```
JLabel lblPWS = new JLabel("密碼 : ");  
lblPWS.setBounds(30, 80, 50, 20);  
frame.add(lblPWS);
```

```
pwfPWS = new JPasswordField();  
pwfPWS.setBounds(70, 80, 120, 20);  
pwfPWS.setEchoChar('*');
```

事件傾聽者

```
frame.add(pwfPWS);
```

```
txtShow = new JTextField();  
txtShow.setBounds(30, 120, 170, 20);  
txtShow.setEditable(false);  
frame.add(txtShow);
```

```
btnOK = new JButton("確定");  
btnOK.setBounds(40, 160, 60, 20);  
btnOK.addActionListener(this);  
frame.add(btnOK);
```

```
btnRST = new JButton("重來");  
btnRST.setBounds(120, 160, 60, 20);  
btnRST.addActionListener(this);  
frame.add(btnRST);
```

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

```
}
```

事件傾聽者

```
public void actionPerformed(ActionEvent e)
{
    if (e.getSource() == btnRST)
    {
        txtID.setText("");
        pwfPWS.setText("");
        txtShow.setText("");
    }

    if (e.getSource() == btnOK)
    {
        String pws = new String(pwfPWS.getPassword());

        if (txtID.getText().equals("MOL") && pws.equals("20170918"))
            txtShow.setText("帳號及密碼正確，歡迎光臨");
        else
            txtShow.setText("帳號及密碼錯誤，請重新輸入");
    }
}
```

事件傾聽者

```
    }  
}  
  
public class CH09_07  
{  
    public static void main(String[] args)  
    {  
        CLoginF loginF = new CLoginF();  
    }  
}
```

事件傾聽者

```
1 package CH09_07;
2
3 import javax.swing.*;
4 import java.awt.event.*;
5
6 class CLoginF extends JFrame implements ActionListener
7 {
8     private JTextField txtID, txtShow;
9     private JPasswordField pwfPWS;
10    private JButton btnOK, btnRST;
11
12    CLoginF()
13    {
14        JFrame frame = new JFrame("登入");
15        frame.setLayout(null);
16        frame.setBounds(100, 100, 235, 230);
17        frame.setVisible(true);
18        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19
20        JLabel lblTitle = new JLabel("請輸入帳號及密碼");
21        lblTitle.setBounds(60, 20, 200, 20);
22        frame.add(lblTitle);
23
24        JLabel lblID = new JLabel("帳號：");
25        lblID.setBounds(30, 50, 50, 20);
26        frame.add(lblID);
```

事件傾聽者

```
27
28     txtID = new JTextField();
29     txtID.setBounds(70, 50, 120, 20);
30     frame.add(txtID);
31
32     JLabel lblPWS = new JLabel("密碼：");
33     lblPWS.setBounds(30, 80, 50, 20);
34     frame.add(lblPWS);
35
36     pwfPWS = new JPasswordField();
37     pwfPWS.setBounds(70, 80, 120, 20);
38     pwfPWS.setEchoChar('*');
39     frame.add(pwfPWS);
40
41     txtShow = new JTextField();
42     txtShow.setBounds(30, 120, 170, 20);
43     txtShow.setEditable(false);
44     frame.add(txtShow);
45
46     btnOK = new JButton("確定");
47     btnOK.setBounds(40, 160, 60, 20);
48     btnOK.addActionListener(this);
49     frame.add(btnOK);
50
51     btnRST = new JButton("重來");
52     btnRST.setBounds(120, 160, 60, 20);
```

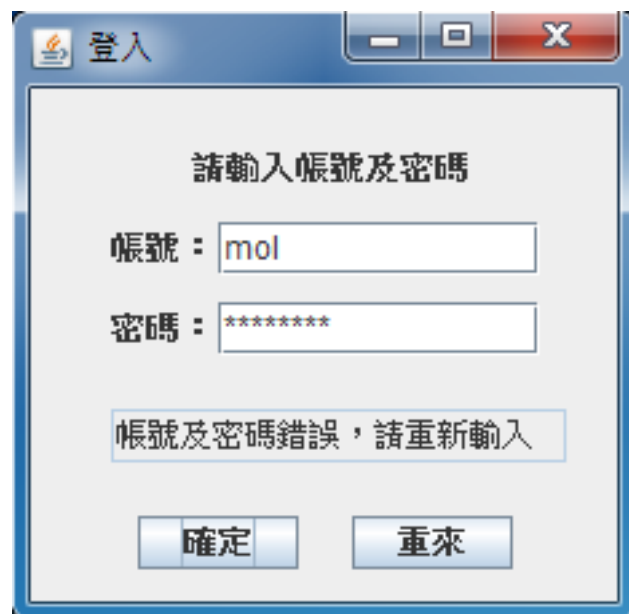

事件傾聽者

```
53     btnRST.addActionListener(this);
54     frame.add(btnRST);
55
56     frame.repaint();
57 }
58
59 public void actionPerformed(ActionEvent e)
60 {
61     if (e.getSource() == btnRST)
62     {
63         txtID.setText("");
64         pwfPWS.setText("");
65         txtShow.setText("");
66     }
67
68     if (e.getSource() == btnOK)
69     {
70         String pws = new String(pwfPWS.getPassword());
71
72         if (txtID.getText().equals("MOL") && pws.equals("20170918"))
73             txtShow.setText("帳號及密碼正確，歡迎光臨");
74         else
75             txtShow.setText("帳號及密碼錯誤，請重新輸入");
76     }
77 }
78 }
```

事件傾聽者

```
79  
80 public class CH09_07  
81 {  
82     public static void main(String[] args)  
83     {  
84         CLoginF loginF = new CLoginF();  
85     }  
86 }
```

◆ 執行結果：



事件傾聽者

◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。
- 行03：
 - ◆ 載入「javax.swing.*」套件。
- 行04：
 - ◆ 載入「java.awt.event.*」套件。
- 行06~行78：
 - ◆ 建立繼承自「JFrame」父類別，並實作「ActionListener」介面的「CLoginF」子類別。
 - ◆ 行08~行10：
 - ◆ 宣告私有欄位成員。
 - ◆ 行12~行57：
 - ◆ 宣告類別「建構子」 「CLoginF()」。

事件傾聽者

- ◆ 行14~行18：
 - ◆ 建立視窗（frame）物件。
- ◆ 行20~行22：
 - ◆ 建立標籤（lblTitle）物件。
- ◆ 行24~行26：
 - ◆ 建立標籤（lblID）物件。
- ◆ 行28~行30：
 - ◆ 建立單行文字欄位（txtID）物件。
- ◆ 行32~行34：
 - ◆ 建立標籤（lblPWS）物件。
- ◆ 行36~行39：
 - ◆ 建立密碼文字欄位（pwfPWS）物件。
- ◆ 行41~行44：
 - ◆ 建立單行文字欄位（txtShow）物件。

事件傾聽者

- ◆ 行46~行49：
 - ◆ 建立按鈕（btnOK）物件。
 - ◆ 行48：

設定事件來源物件連結事件傾聽者。

this是指CLoginF類別所建立的視窗物件。
- ◆ 行51~行54：
 - ◆ 建立按鈕（btnRST）物件。
 - ◆ 行53：

設定事件來源物件連結事件傾聽者。

this是指CLoginF類別所建立的視窗物件。
- ◆ 行56：
 - ◆ 重繪視窗元件。

事件傾聽者

- ◆ 行59~行77：

- ◆ 實作ActionListener介面的actionPerformed()方法。
- ◆ 為觸發事件時的處理方法。
- ◆ ActionEvent被觸發時，e就是事件的來源。

- ◆ 行61~行66：

當e.getSource()取得的事件來源物件為btnRST時，執行此區段程式。

用setText()方法分別設定txtID、pwfPWS、txtShow為空字串。

- ◆ 行68~行76：

當e.getSource()取得的事件來源物件為btnOK時，便會執行此區段程式。

行70：

用getPassword()方法取得pwfPWS欄位資料，再轉為字串物件。

事件傾聽者

行72：

用getText()方法取得txtID欄位資料，並與字串「MOL」比對是否相等，同時檢察密碼是否為「20170918」。

行73：

若帳號與密碼相同則用setText()方法，設定txtShow欄位顯示「帳號及密碼正確，歡迎光臨」。

行75：

若帳號或密碼比對不相同，則txtShow欄位顯示為「帳號及密碼錯誤，請重新輸入」。

- 行84：

- ◆ 利用「CLoginF()」的「建構子」，建立loginF物件。

事件傾聽者

◆ 程式（以「元件」為傾聽對象）：

```
package CH09_08;
```

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

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

```
class CLoginF extends JFrame
```

```
{
```

```
    private JTextField txtID, txtShow;
```

```
    private JPasswordField pwfPWS;
```

```
    private JButton btnOK, btnRST;
```

```
    CLoginF()
```

```
{
```

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

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

```
        frame.setBounds(100, 100, 235, 230);
```

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

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


事件傾聽者

```
JLabel lblTitle = new JLabel("請輸入帳號及密碼");  
lblTitle.setBounds(60, 20, 200, 20);  
frame.add(lblTitle);
```

```
JLabel lblID = new JLabel("帳號 : ");  
lblID.setBounds(30, 50, 50, 20);  
frame.add(lblID);
```

```
txtID = new JTextField();  
txtID.setBounds(70, 50, 120, 20);  
frame.add(txtID);
```

```
JLabel lblPWS = new JLabel("密碼 : ");  
lblPWS.setBounds(30, 80, 50, 20);  
frame.add(lblPWS);
```

```
pwfPWS = new JPasswordField();  
pwfPWS.setBounds(70, 80, 120, 20);  
pwfPWS.setEchoChar('*');
```

事件傾聽者

```
frame.add(pwfPWS);
```

```
txtShow = new JTextField();  
txtShow.setBounds(30, 120, 170, 20);  
txtShow.setEditable(false);  
frame.add(txtShow);
```

```
btnOK = new JButton("確定");  
btnOK.setBounds(40, 160, 60, 20);  
btnOK.addActionListener(listenerbtn);  
frame.add(btnOK);
```

```
btnRST = new JButton("重來");  
btnRST.setBounds(120, 160, 60, 20);  
btnRST.addActionListener(listenerbtn);  
frame.add(btnRST);
```

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

```
}
```

事件傾聽者

```
public ActionListener Listenerbtn=new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource() == btnRST)
        {
            txtID.setText("");
            pwfPWS.setText("");
            txtShow.setText("");
        }

        if (e.getSource() == btnOK)
        {
            String pws = new String(pwfPWS.getPassword());

            if (txtID.getText().equals("MOL") && pws.equals("20170918"))
                txtShow.setText("帳號及密碼正確，歡迎光臨");
            else
                txtShow.setText("帳號及密碼錯誤，請重新輸入");
        }
    }
}
```

事件傾聽者

```
    }  
    }  
};  
  
public class CH09_08  
{  
    public static void main(String[] args)  
    {  
        CLoginF loginF = new CLoginF();  
    }  
}
```

事件傾聽者

```
1 package CH09_08;
2
3 import javax.swing.*;
4 import java.awt.event.*;
5
6 class CLoginF extends JFrame
7 {
8     private JTextField txtID, txtShow;
9     private JPasswordField pwfPWS;
10    private JButton btnOK, btnRST;
11
12    CLoginF()
13    {
14        JFrame frame = new JFrame("登入");
15        frame.setLayout(null);
16        frame.setBounds(100, 100, 235, 230);
17        frame.setVisible(true);
18        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19
20        JLabel lblTitle = new JLabel("請輸入帳號及密碼");
21        lblTitle.setBounds(60, 20, 200, 20);
22        frame.add(lblTitle);
23
24        JLabel lblID = new JLabel("帳號：");
25        lblID.setBounds(30, 50, 50, 20);
26        frame.add(lblID);
```

事件傾聽者

```
27
28     txtID = new JTextField();
29     txtID.setBounds(70, 50, 120, 20);
30     frame.add(txtID);
31
32     JLabel lblPWS = new JLabel("密碼：");
33     lblPWS.setBounds(30, 80, 50, 20);
34     frame.add(lblPWS);
35
36     pwfPWS = new JPasswordField();
37     pwfPWS.setBounds(70, 80, 120, 20);
38     pwfPWS.setEchoChar('*');
39     frame.add(pwfPWS);
40
41     txtShow = new JTextField();
42     txtShow.setBounds(30, 120, 170, 20);
43     txtShow.setEditable(false);
44     frame.add(txtShow);
45
46     btnOK = new JButton("確定");
47     btnOK.setBounds(40, 160, 60, 20);
48     btnOK.addActionListener(listenerbtn);
49     frame.add(btnOK);
50
51     btnRST = new JButton("重來");
52     btnRST.setBounds(120, 160, 60, 20);
```

事件傾聽者

```
53     btnRST.addActionListener(listenerbtn);
54     frame.add(btnRST);
55
56     frame.repaint();
57 }
58
59 public ActionListener listenerbtn=new ActionListener()
60 {
61     public void actionPerformed(ActionEvent e)
62     {
63         if (e.getSource() == btnRST)
64         {
65             txtID.setText("");
66             pwfPWS.setText("");
67             txtShow.setText("");
68         }
69
70         if (e.getSource() == btnOK)
71         {
72             String pws = new String(pwfPWS.getPassword());
73
74             if (txtID.getText().equals("MOL") && pws.equals("20170918"))
75                 txtShow.setText("帳號及密碼正確，歡迎光臨");
76             else
77                 txtShow.setText("帳號及密碼錯誤，請重新輸入");
78         }
79     }
80 }
```

事件傾聽者

```
79     }  
80   };  
81 }  
82  
83 public class CH09_08  
84 {  
85     public static void main(String[] args)  
86     {  
87         CLoginF loginF = new CLoginF();  
88     }  
89 }
```

◆ 執行結果：



事件傾聽者

◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。
- 行03：
 - ◆ 載入「javax.swing.*」套件。
- 行04：
 - ◆ 載入「java.awt.event.*」套件。
- 行06~行81：
 - ◆ 建立繼承自「JFrame」父類別的「CLoginF」子類別。
 - ◆ 行08~行10：
 - ◆ 宣告私有欄位成員。
 - ◆ 行12~行57：
 - ◆ 宣告類別「建構子」「CLoginF()」。

事件傾聽者

- ◆ 行14~行18：
 - ◆ 建立視窗（frame）物件。
- ◆ 行20~行22：
 - ◆ 建立標籤（lblTitle）物件。
- ◆ 行24~行26：
 - ◆ 建立標籤（lblID）物件。
- ◆ 行28~行30：
 - ◆ 建立單行文字欄位（txtID）物件。
- ◆ 行32~行34：
 - ◆ 建立標籤（lblPWS）物件。
- ◆ 行36~行39：
 - ◆ 建立密碼文字欄位（pwfPWS）物件。
- ◆ 行41~行44：
 - ◆ 建立單行文字欄位（txtShow）物件。

事件傾聽者

- ◆ 行46~行49：
 - ◆ 建立按鈕（btnOK）物件。
 - ◆ 行48：

設定事件來源物件連結事件傾聽者。

Listenerbtn是指自訂的傾聽者物件。
- ◆ 行51~行54：
 - ◆ 建立按鈕（btnRST）物件。
 - ◆ 行53：

設定事件來源物件連結事件傾聽者。

Listenerbtn是指自訂的傾聽者物件。
- ◆ 行56：
 - ◆ 重繪視窗元件。

事件傾聽者

- ◆ 行59~行80：

- ◆ 建立傾聽者物件Listenerbtn。

- ◆ 行61~行78：

- 建立actionPerformed()方法。

- 為觸發事件時的處理方法。

- ActionEvent被觸發時，e就是事件的來源。

- 行63~行68：

- 當e.getSource()取得的事件來源物件為btnRST時，執行此區段程式。

- 用setText()方法分別設定txtID、pwfPWS、txtShow為空字串。

- 行70~行78：

- 當e.getSource()取得的事件來源物件為btnOK時，便會執行此區段程式。

事件傾聽者

行72：

用 getPassword() 方法取得 pwfPWS 欄位資料，再轉為字串物件。

行74：

用 getText() 方法取得 txtID 欄位資料，並與字串「MOL」比對是否相等，同時檢察密碼是否為「20170918」。

行75：

若帳號與密碼相同則用 setText() 方法，設定 txtShow 欄位顯示「帳號及密碼正確，歡迎光臨」。

行77：

若帳號或密碼比對不相同，則 txtShow 欄位顯示為「帳號及密碼錯誤，請重新輸入」。

- 行87：

- ◆ 利用「CLoginF()」的「建構子」，建立 loginF 物件。

事件傾聽者

◆ 程式（匿名的「事件傾聽者」）：

```
package CH09_09;
```

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

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

```
class CLoginF extends JFrame
```

```
{
```

```
    private JTextField txtID, txtShow;
```

```
    private JPasswordField pwfPWS;
```

```
    private JButton btnOK, btnRST;
```

```
    CLoginF()
```

```
{
```

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

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

```
        frame.setBounds(100, 100, 235, 230);
```

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

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

事件傾聽者

```
JLabel lblTitle = new JLabel("請輸入帳號及密碼");  
lblTitle.setBounds(60, 20, 200, 20);  
frame.add(lblTitle);
```

```
JLabel lblID = new JLabel("帳號 : ");  
lblID.setBounds(30, 50, 50, 20);  
frame.add(lblID);
```

```
txtID = new JTextField();  
txtID.setBounds(70, 50, 120, 20);  
frame.add(txtID);
```

```
JLabel lblPWS = new JLabel("密碼 : ");  
lblPWS.setBounds(30, 80, 50, 20);  
frame.add(lblPWS);
```

```
pwfPWS = new JPasswordField();  
pwfPWS.setBounds(70, 80, 120, 20);  
pwfPWS.setEchoChar('*');
```

事件傾聽者

```
frame.add(pwfPWS);

txtShow = new JTextField();
txtShow.setBounds(30, 120, 170, 20);
txtShow.setEditable(false);
frame.add(txtShow);

btnOK = new JButton("確定");
btnOK.setBounds(40, 160, 60, 20);
btnOK.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        String pws = new String(pwfPWS.getPassword());

        if (txtID.getText().equals("MOL") && pws.equals("20170918"))
            txtShow.setText("帳號及密碼正確，歡迎光臨");
        else
            txtShow.setText("帳號及密碼錯誤，請重新輸入");
    }
});
```


事件傾聽者

```
});  
frame.add(btnOK);  
  
btnRST = new JButton("重來");  
btnRST.setBounds(120, 160, 60, 20);  
btnRST.addActionListener(new ActionListener()  
{  
    public void actionPerformed(ActionEvent e)  
    {  
        txtID.setText("");  
        pwfPWS.setText("");  
        txtShow.setText("");  
    }  
});  
frame.add(btnRST);  
  
frame.repaint();  
}  
}
```

事件傾聽者

```
public class CH09_09
{
    public static void main(String[] args)
    {
        CLoginF loginF = new CLoginF();
    }
}
```

事件傾聽者

```
1 package CH09_09;
2
3 import javax.swing.*;
4 import java.awt.event.*;
5
6 class CLoginF extends JFrame
7 {
8     private JTextField txtID, txtShow;
9     private JPasswordField pwfPWS;
10    private JButton btnOK, btnRST;
11
12    CLoginF()
13    {
14        JFrame frame = new JFrame("登入");
15        frame.setLayout(null);
16        frame.setBounds(100, 100, 235, 230);
17        frame.setVisible(true);
18        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19
20        JLabel lblTitle = new JLabel("請輸入帳號及密碼");
21        lblTitle.setBounds(60, 20, 200, 20);
22        frame.add(lblTitle);
23
24        JLabel lblID = new JLabel("帳號 : ");
25        lblID.setBounds(30, 50, 50, 20);
26        frame.add(lblID);
```

事件傾聽者

```
27
28     txtID = new JTextField();
29     txtID.setBounds(70, 50, 120, 20);
30     frame.add(txtID);
31
32     JLabel lblPWS = new JLabel("密碼：");
33     lblPWS.setBounds(30, 80, 50, 20);
34     frame.add(lblPWS);
35
36     pwfPWS = new JPasswordField();
37     pwfPWS.setBounds(70, 80, 120, 20);
38     pwfPWS.setEchoChar('*');
39     frame.add(pwfPWS);
40
41     txtShow = new JTextField();
42     txtShow.setBounds(30, 120, 170, 20);
43     txtShow.setEditable(false);
44     frame.add(txtShow);
45
46     btnOK = new JButton("確定");
47     btnOK.setBounds(40, 160, 60, 20);
48     btnOK.addActionListener(new ActionListener()
49     {
50         public void actionPerformed(ActionEvent e)
51         {
52             String pws = new String(pwfPWS.getPassword());
```

事件傾聽者

```
53
54         if (txtID.getText().equals("MOL") && pws.equals("20170918"))
55             txtShow.setText("帳號及密碼正確，歡迎光臨");
56         else
57             txtShow.setText("帳號及密碼錯誤，請重新輸入");
58     }
59 });
60 frame.add(btnOK);
61
62 btnRST = new JButton("重來");
63 btnRST.setBounds(120, 160, 60, 20);
64 btnRST.addActionListener(new ActionListener()
65 {
66     public void actionPerformed(ActionEvent e)
67     {
68         txtID.setText("");
69         pwfPWS.setText("");
70         txtShow.setText("");
71     }
72 });
73 frame.add(btnRST);
74
75 frame.repaint();
76 }
77 }
78
```

事件傾聽者

```
79 public class CH09_09
80 {
81     public static void main(String[] args)
82     {
83         CLoginF loginF = new CLoginF();
84     }
85 }
```

◆ 執行結果：



事件傾聽者

◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。
- 行03：
 - ◆ 載入「javax.swing.*」套件。
- 行04：
 - ◆ 載入「java.awt.event.*」套件。
- 行06~行77：
 - ◆ 建立繼承自「JFrame」父類別的「CLoginF」子類別。
 - ◆ 行08~行10：
 - ◆ 宣告私有欄位成員。
 - ◆ 行12~行76：
 - ◆ 宣告類別「建構子」「CLoginF()」。

事件傾聽者

- ◆ 行14~行18：
 - ◆ 建立視窗（frame）物件。
- ◆ 行20~行22：
 - ◆ 建立標籤（lblTitle）物件。
- ◆ 行24~行26：
 - ◆ 建立標籤（lblID）物件。
- ◆ 行28~行30：
 - ◆ 建立單行文字欄位（txtID）物件。
- ◆ 行32~行34：
 - ◆ 建立標籤（lblPWS）物件。
- ◆ 行36~行39：
 - ◆ 建立密碼文字欄位（pwfPWS）物件。
- ◆ 行41~行44：
 - ◆ 建立單行文字欄位（txtShow）物件。

事件傾聽者

- ◆ 行46~行60：

- ◆ 建立按鈕（btnOK）物件。

- ◆ 行48~行59：

- 設定事件來源物件連結事件傾聽者。

- 此為匿名傾聽者物件（在物件中，以new關鍵字建立傾聽者物件，並直接定義actionPerformed()方法）。

- 行50~行58：

- 建立actionPerformed()方法。

- 行52：

- 用getPassword()方法取得pwfPWS欄位資料，再轉為字串物件。

- 行54：

- 用getText()方法取得txtID欄位資料，並與字串「MOL」比對是否相等，同時檢察密碼是否為「20170918」。

事件傾聽者

行55：

若帳號與密碼相同則用setText()方法，設定txtShow欄位顯示「帳號及密碼正確，歡迎光臨」。

行57：

若帳號或密碼比對不相同，則txtShow欄位顯示為「帳號及密碼錯誤，請重新輸入」。

◆ 行62~行73：

◆ 建立按鈕（btnRST）物件。

◆ 行64：

設定事件來源物件連結事件傾聽者。

此為匿名傾聽者物件（在物件中，以new關鍵字建立傾聽者物件，並直接定義actionPerformed()方法）。

事件傾聽者

行66~行71：

建立actionPerformed()方法。

行68~行70：

用setText()方法分別設定txtID、pwfPWS、
txtShow為空字串。

- ◆ 行75：

- ◆ 重繪視窗元件。

- 行83：

- ◆ 利用「CLoginF()」的「建構子」，建立loginF物件。

取消註冊事件傾聽者

- 在某些情況下，已註冊事件傾聽者的來源物件必須要取消傾聽者的註冊，其語法為：

事件來源物件.removeActionListener(事件傾聽者);

- 被註冊為傾聽者的一定是物件，而且要命名才能有被取消的功能。

例如：

```
btnOK.addActionListener(this);
```

➤ this是視窗指物件。

或

```
btnRST.addActionListener(Listenerbtn);
```

➤ Listenerbtn是自訂的傾聽者物件。

取消註冊事件傾聽者

■ 若被註冊的傾聽者物件，是匿名的傾聽者物件，則無法被取消。

例如：

```
btnOK.addActionListener(new ActionListener());
```

取消註冊事件傾聽者

◆ 程式（以「視窗」為傾聽對象）：

```
package CH09_10;
```

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

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

```
class CAreaF extends JFrame implements ActionListener
```

```
{
```

```
    private JTextField txtLen = new JTextField("0");
```

```
    private JTextField txtArea = new JTextField();
```

```
    private JButton btnOK = new JButton("確定");
```

```
    private JButton btnCls = new JButton("清除");
```

```
    CAreaF()
```

```
{
```

```
        setTitle("正方型面積");
```

```
        setLayout(null);
```

```
        setBounds(100, 100, 220, 180);
```

```
        setVisible(true);
```

取消註冊事件傾聽者

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

```
JLabel lblTitle = new JLabel("正方形面積 = L * L");  
lblTitle.setBounds(50, 10, 200, 20);  
add(lblTitle);
```

```
JLabel lblLen = new JLabel("輸入邊長 : ");  
lblLen.setBounds(20, 50, 100, 20);  
add(lblLen);
```

```
txtLen.setBounds(90, 50, 100, 20);  
add(txtLen);
```

```
txtArea.setBounds(20, 80, 170, 20);  
txtArea.setEditable(false);  
add(txtArea);
```

```
btnOK.setBounds(20, 110, 70, 20);  
add(btnOK);  
btnOK.addActionListener(this);
```

取消註冊事件傾聽者

```
btnCls.setBounds(120, 110, 70, 20);  
add(btnCls);  
  
repaint();  
}  
  
public void actionPerformed(ActionEvent e)  
{  
    if (e.getSource() == btnOK)  
    {  
        int len, area;  
        len = Integer.parseInt(txtLen.getText());  
        area = len * len;  
        txtArea.setText("面積： " + area);  
        btnOK.removeActionListener(this);  
        btnCls.addActionListener(this);  
    }  
  
    if (e.getSource() == btnCls)
```


取消註冊事件傾聽者

```
{
    txtLen.setText("0");
    txtArea.setText(" ");
    btnCls.removeActionListener(this);
    btnOK.addActionListener(this);
}
}
```

```
public class CH09_10
{
    public static void main(String[] args)
    {
        new CAreaF();
    }
}
```

取消註冊事件傾聽者

```
1 package CH09_10;
2
3 import javax.swing.*;
4 import java.awt.event.*;
5
6 class CAreaF extends JFrame implements ActionListener
7 {
8     private JTextField txtLen = new JTextField("0");
9     private JTextField txtArea = new JTextField();
10    private JButton btnOK = new JButton("確定");
11    private JButton btnCls = new JButton("清除");
12
13    CAreaF()
14    {
15        setTitle("正方形面積");
16        setLayout(null);
17        setBounds(100, 100, 220, 180);
18        setVisible(true);
19        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20
21        JLabel lblTitle = new JLabel("正方形面積 = L * L");
22        lblTitle.setBounds(50, 10, 200, 20);
23        add(lblTitle);
24
25        JLabel lblLen = new JLabel("輸入邊長：");
26        lblLen.setBounds(20, 50, 100, 20);
```

取消註冊事件傾聽者

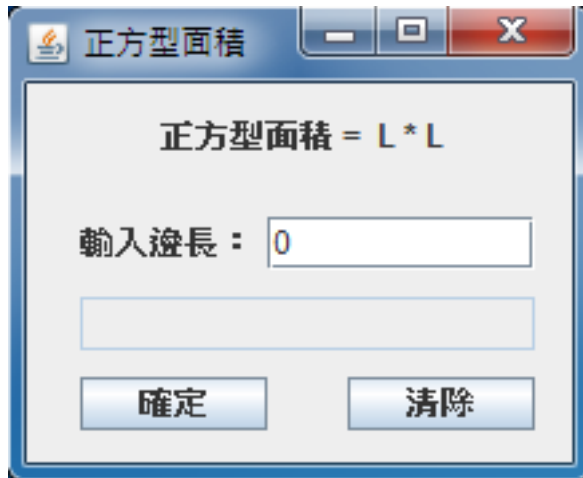
```
27     add(lblLen);
28
29     txtLen.setBounds(90, 50, 100, 20);
30     add(txtLen);
31
32     txtArea.setBounds(20, 80, 170, 20);
33     txtArea.setEditable(false);
34     add(txtArea);
35
36     btnOK.setBounds(20, 110, 70, 20);
37     add(btnOK);
38     btnOK.addActionListener(this);
39
40     btnCls.setBounds(120, 110, 70, 20);
41     add(btnCls);
42
43     repaint();
44 }
45
46 public void actionPerformed(ActionEvent e)
47 {
48     if (e.getSource() == btnOK)
49     {
50         int len, area;
51         len = Integer.parseInt(txtLen.getText());
52         area = len * len;
```

取消註冊事件傾聽者

```
53         txtArea.setText("面積： " + area);
54         btnOK.removeActionListener(this);
55         btnCls.addActionListener(this);
56     }
57
58     if (e.getSource() == btnCls)
59     {
60         txtLen.setText("0");
61         txtArea.setText(" ");
62         btnCls.removeActionListener(this);
63         btnOK.addActionListener(this);
64     }
65 }
66
67
68 public class CH09_10
69 {
70     public static void main(String[] args)
71     {
72         new CAreaF0();
73     }
74 }
```

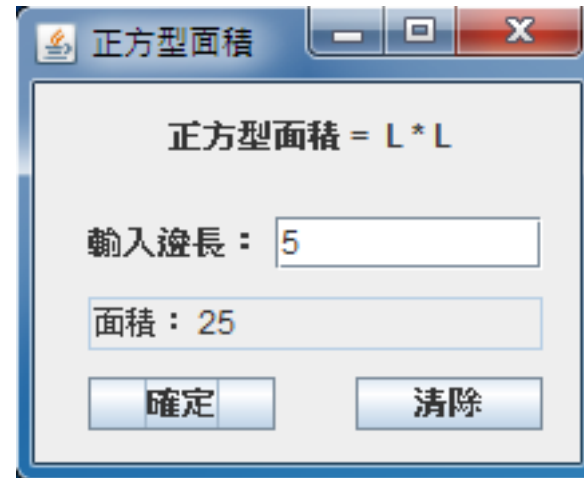
取消註冊事件傾聽者

◆ 執行結果：



正方形面積 = $L * L$

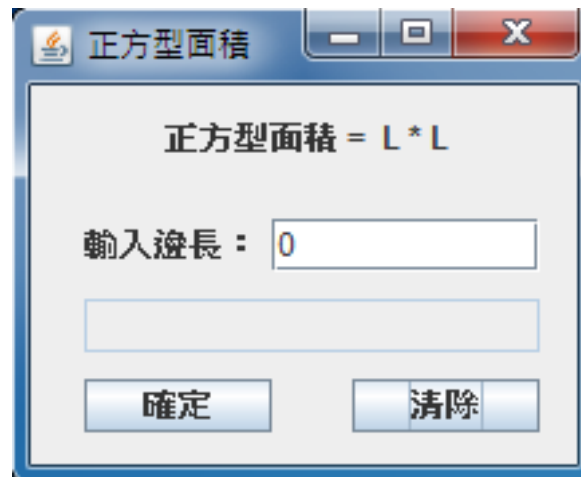
輸入邊長：



正方形面積 = $L * L$

輸入邊長：

面積：



正方形面積 = $L * L$

輸入邊長：

取消註冊事件傾聽者

◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。
- 行03：
 - ◆ 載入「javax.swing.*」套件。
- 行04：
 - ◆ 載入「java.awt.event.*」套件。
- 行06~行66：
 - ◆ 建立繼承自「JFrame」父類別，並實作「ActionListener」介面的「CAreaF」子類別。
 - ◆ 行08~行11：
 - ◆ 宣告私有物件成員。
 - ◆ 行13~行44：
 - ◆ 宣告類別「建構子」「CAreaF」。

取消註冊事件傾聽者

- ◆ 行15~行19：
 - ◆ 建立視窗物件。
- ◆ 行21~行23：
 - ◆ 建立標籤（lblTitle）物件。
- ◆ 行25~行27：
 - ◆ 建立標籤（lblLen）物件。
- ◆ 行29~行30：
 - ◆ 設定單行文字欄位（txtLen）物件。
- ◆ 行32~行34：
 - ◆ 設定單行文字欄位（txtArea）物件。
- ◆ 行36~行38：
 - ◆ 設定按鈕（btnOK）物件。
 - ◆ 行38：
 - 設定事件來源物件連結事件傾聽者。
 - this是指CAreaF類別所建立的視窗物件。

取消註冊事件傾聽者

- ◆ 行40~行41：
 - ◆ 建立按鈕（btnCls）物件。
- ◆ 行43：
 - ◆ 重繪視窗元件。
- ◆ 行46~行65：
 - ◆ 實作ActionListener介面的actionPerformed()方法。
 - ◆ 為觸發事件時的處理方法。
 - ◆ ActionEvent被觸發時，e就是事件的來源。
 - ◆ 行48~行56：

當e.getSource()取得的事件來源物件為btnOK時，執行此區段程式。

計算面積，並用setText()方法，將計算結果寫入txtArea中。
 - 行54：

取消註冊btnOK事件傾聽者。

取消註冊事件傾聽者

行55：

設定btnCls連結事件傾聽者。

this是指CAreaF類別所建立的視窗物件。

行58~行64：

當e.getSource()取得的事件來源物件為btnCls時，執行此區段程式。

用setText()方法設定txtLen、txtArea的值。

行62：

取消註冊btnCls事件傾聽者。

行63：

設定btnOK連結事件傾聽者。

this是指CAreaF類別的視窗物件。

- 行72：

- ◆ 利用「CAreaF()」的「建構子」，建立視窗物件。
- ◆ 因為視窗物件未命名宣告，所以它是一種「匿名物件」。

取消註冊事件傾聽者

注意：

因為此視窗為匿名物件，故無法在main()中，重新設定視窗的大小、位置...等屬性值。

取消註冊事件傾聽者

◆ 程式（以「元件」為傾聽對象）：

```
package CH09_11;
```

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

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

```
class CAreaF extends JFrame
```

```
{
```

```
    private JTextField txtLen, txtArea;
```

```
    private JButton btnOK, btnCls;
```

```
    CAreaF()
```

```
{
```

```
    setTitle("正方形面積");
```

```
    setLayout(null);
```

```
    setBounds(100, 100, 220, 180);
```

```
    setVisible(true);
```

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

取消註冊事件傾聽者

```
JLabel lblTitle = new JLabel("正方型面積 = L * L");  
lblTitle.setBounds(50, 10, 200, 20);  
add(lblTitle);
```

```
JLabel lblLen = new JLabel("輸入邊長 : ");  
lblLen.setBounds(20, 50, 100, 20);  
add(lblLen);
```

```
txtLen=new JTextField("0");  
txtLen.setBounds(90, 50, 100, 20);  
add(txtLen);
```

```
txtArea=new JTextField();  
txtArea.setBounds(20, 80, 170, 20);  
txtArea.setEditable(false);  
add(txtArea);
```

```
btnOK=new JButton("確定");  
btnOK.setBounds(20, 110, 70, 20);
```

取消註冊事件傾聽者

```
add(btnOK);  
btnOK.addActionListener(listenerOK);  
  
btnCls=new JButton("清除");  
btnCls.setBounds(120, 110, 70, 20);  
add(btnCls);  
  
repaint();  
}  
  
public ActionListener listenerOK=new ActionListener()  
{  
    public void actionPerformed(ActionEvent e)  
    {  
        int len, area;  
        len = Integer.parseInt(txtLen.getText());  
        area = len * len;  
        txtArea.setText("面積： " + area);  
        btnOK.removeActionListener(listenerOK);  
        btnCls.addActionListener(listenerCls);  
    }  
}
```

取消註冊事件傾聽者

```
    }  
    };  
  
    public ActionListener ListenerCls=new ActionListener()  
    {  
        public void actionPerformed(ActionEvent e)  
        {  
            txtLen.setText("0");  
            txtArea.setText(" ");  
            btnCls.removeActionListener(ListenerCls);  
            btnOK.addActionListener(ListenerOK);  
        }  
    };  
}  
  
public class CH09_11  
{  
    public static void main(String[] args)  
    {  
        CAreaF areaF=new CAreaF();  
    }  
}
```

取消註冊事件傾聽者

```
}  
}
```

取消註冊事件傾聽者

```
1 package CH09_11;
2
3 import javax.swing.*;
4 import java.awt.event.*;
5
6 class CAreaF extends JFrame
7 {
8     private JTextField txtLen, txtArea;
9     private JButton btnOK, btnCls;
10
11     CAreaF()
12     {
13         setTitle("正方型面積");
14         setLayout(null);
15         setBounds(100, 100, 220, 180);
16         setVisible(true);
17         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18
19         JLabel lblTitle = new JLabel("正方型面積 = L * L");
20         lblTitle.setBounds(50, 10, 200, 20);
21         add(lblTitle);
22
23         JLabel lblLen = new JLabel("輸入邊長：");
24         lblLen.setBounds(20, 50, 100, 20);
25         add(lblLen);
26
```


取消註冊事件傾聽者

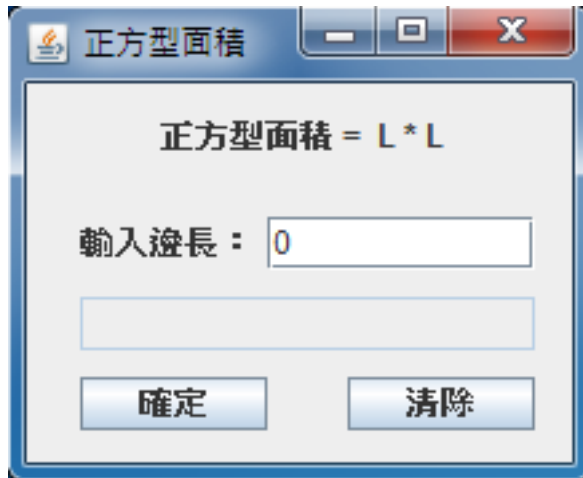
```
27     txtLen=new JTextField("0");
28     txtLen.setBounds(90, 50, 100, 20);
29     add(txtLen);
30
31     txtArea=new JTextField();
32     txtArea.setBounds(20, 80, 170, 20);
33     txtArea.setEditable(false);
34     add(txtArea);
35
36     btnOK=new JButton("確定");
37     btnOK.setBounds(20, 110, 70, 20);
38     add(btnOK);
39     btnOK.addActionListener(listenerOK);
40
41     btnCls=new JButton("清除");
42     btnCls.setBounds(120, 110, 70, 20);
43     add(btnCls);
44
45     repaint();
46 }
47
48 public ActionListener listenerOK=new ActionListener()
49 {
50     public void actionPerformed(ActionEvent e)
51     {
52         int len, area;
```

取消註冊事件傾聽者

```
53         len = Integer.parseInt(txtLen.getText());
54         area = len * len;
55         txtArea.setText("面積： " + area);
56         btnOK.removeActionListener(listenerOK);
57         btnCls.addActionListener(listenerCls);
58     }
59 };
60
61 public ActionListener listenerCls=new ActionListener()
62 {
63     public void actionPerformed(ActionEvent e)
64     {
65         txtLen.setText("0");
66         txtArea.setText(" ");
67         btnCls.removeActionListener(listenerCls);
68         btnOK.addActionListener(listenerOK);
69     }
70 };
71 }
72
73 public class CH09_11
74 {
75     public static void main(String[] args)
76     {
77         CAreaF areaF=new CAreaF();
78     }
79 }
```

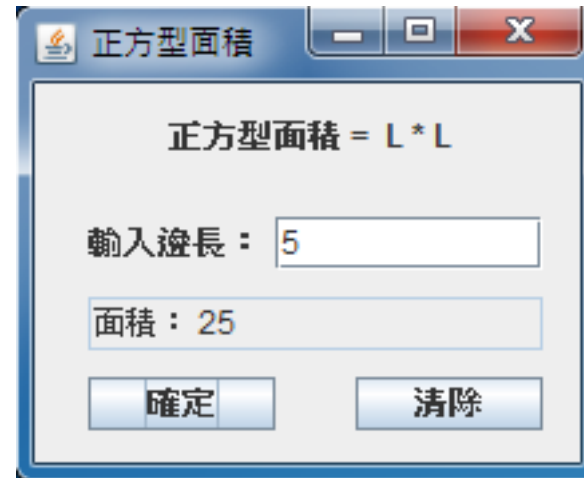
取消註冊事件傾聽者

◆ 執行結果：



正方形面積 = $L * L$

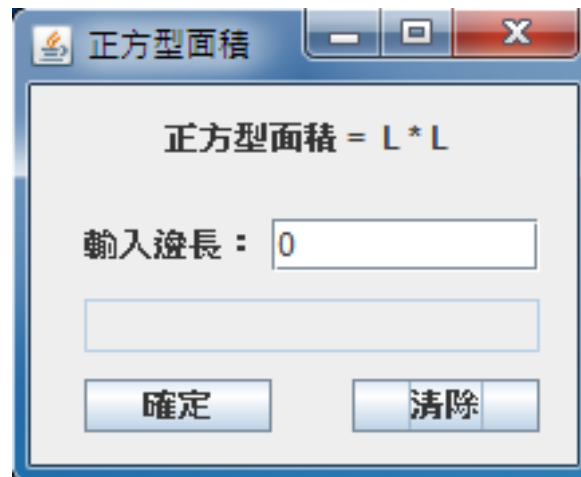
輸入邊長：



正方形面積 = $L * L$

輸入邊長：

面積：



正方形面積 = $L * L$

輸入邊長：

取消註冊事件傾聽者

◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。
- 行03：
 - ◆ 載入「javax.swing.*」套件。
- 行04：
 - ◆ 載入「java.awt.event.*」套件。
- 行06~行71：
 - ◆ 建立繼承自「JFrame」父類別的「CAreaF」子類別。
 - ◆ 行08~行09：
 - ◆ 宣告私有欄位成員。
 - ◆ 行11~行46：
 - ◆ 宣告類別「建構子」「CAreaF」。

取消註冊事件傾聽者

- ◆ 行13~行17：
 - ◆ 建立視窗物件。
- ◆ 行19~行21：
 - ◆ 建立標籤（lblTitle）物件。
- ◆ 行23~行25：
 - ◆ 建立標籤（lblLen）物件。
- ◆ 行27~行29：
 - ◆ 建立單行文字欄位（txtLen）物件。
- ◆ 行31~行34：
 - ◆ 建立單行文字欄位（txtArea）物件。
- ◆ 行36~行39：
 - ◆ 建立按鈕（btnOK）物件。
 - ◆ 行39：
 - 設定事件來源物件連結事件傾聽者。
 - ListenerOK是指自訂的傾聽者物件。

取消註冊事件傾聽者

- ◆ 行41~行43：
 - ◆ 建立按鈕（btnCls）物件。
- ◆ 行45：
 - ◆ 重繪視窗元件。
- ◆ 行48~行59：
 - ◆ 建立ListenerOK傾聽者物件。
 - ◆ 行50~行58：
 - 定義actionPerformed()方法。
 - 為觸發事件時的處理方法。
 - ActionEvent被觸發時，e就是事件的來源。
 - 行52~行57：
 - 計算面積，並用setText()方法，將計算結果寫入txtArea中。
 - 行56：
 - 取消註冊ListenerOK事件傾聽者。

取消註冊事件傾聽者

行57：

設定btnCls連結事件傾聽者。

ListenerCls是指自訂的傾聽者物件。

◆ 行61~行70：

◆ 建立ListenerCls傾聽者物件。

◆ 行63~行69：

定義actionPerformed()方法。

為觸發事件時的處理方法。

ActionEvent被觸發時，e就是事件的來源。

行65~行66：

用setText()方法分別設定txtLen、txtArea的值。

行67：

取消註冊ListenerCls事件傾聽者。

取消註冊事件傾聽者

行68：

設定btnOK連結事件傾聽者。

ListenerOK是指自訂的傾聽者物件。

- 行77：

- ◆ 利用「CAreaF()」的「建構子」，建立areaF物件。

取消註冊事件傾聽者

◆ 程式（匿名的「事件傾聽者」）：

```
package CH09_12;
```

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

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

```
class CAreaF extends JFrame
```

```
{
```

```
    private JTextField txtLen, txtArea;
```

```
    private JButton btnOK, btnCls;
```

```
    CAreaF()
```

```
{
```

```
    setTitle("正方形面積");
```

```
    setLayout(null);
```

```
    setBounds(100, 100, 220, 180);
```

```
    setVisible(true);
```

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

取消註冊事件傾聽者

```
JLabel lblTitle = new JLabel("正方型面積 = L * L");  
lblTitle.setBounds(50, 10, 200, 20);  
add(lblTitle);
```

```
JLabel lblLen = new JLabel("輸入邊長 : ");  
lblLen.setBounds(20, 50, 100, 20);  
add(lblLen);
```

```
txtLen = new JTextField("0");  
txtLen.setBounds(90, 50, 100, 20);  
add(txtLen);
```

```
txtArea = new JTextField();  
txtArea.setBounds(20, 80, 170, 20);  
txtArea.setEditable(false);  
add(txtArea);
```

```
btnOK = new JButton("確定");  
btnOK.setBounds(20, 110, 70, 20);
```

取消註冊事件傾聽者

```
add(btnOK);
btnOK.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        int len, area;
        len = Integer.parseInt(txtLen.getText());
        area = len * len;
        txtArea.setText("面積： " + area);
    }
});

btnCls = new JButton("清除");
btnCls.setBounds(120, 110, 70, 20);
add(btnCls);
btnCls.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        txtLen.setText("0");
    }
});
```

取消註冊事件傾聽者

```
        txtArea.setText("");
    }
});

repaint();
}
}

public class CH09_12
{
    public static void main(String[] args)
    {
        CAreaF areaF = new CAreaF();
    }
}
```

取消註冊事件傾聽者

```
1 package CH09_12;
2
3 import javax.swing.*;
4 import java.awt.event.*;
5
6 class CAreaF extends JFrame
7 {
8     private JTextField txtLen, txtArea;
9     private JButton btnOK, btnCls;
10
11     CAreaF()
12     {
13         setTitle("正方型面積");
14         setLayout(null);
15         setBounds(100, 100, 220, 180);
16         setVisible(true);
17         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18
19         JLabel lblTitle = new JLabel("正方型面積 = L * L");
20         lblTitle.setBounds(50, 10, 200, 20);
21         add(lblTitle);
22
23         JLabel lblLen = new JLabel("輸入邊長：");
24         lblLen.setBounds(20, 50, 100, 20);
25         add(lblLen);
26     }
27 }
```

取消註冊事件傾聽者

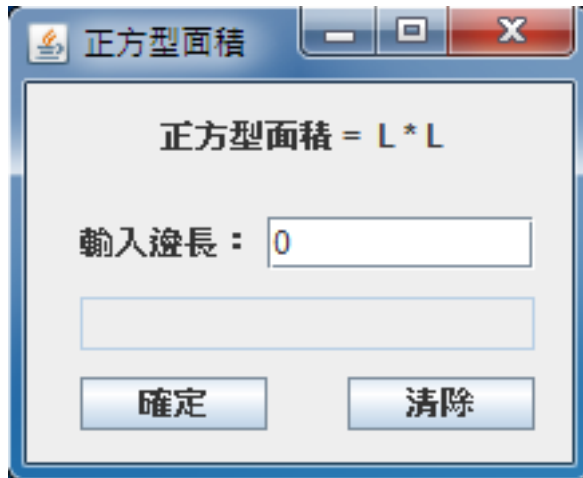
```
27     txtLen = new JTextField("0");
28     txtLen.setBounds(90, 50, 100, 20);
29     add(txtLen);
30
31     txtArea = new JTextField();
32     txtArea.setBounds(20, 80, 170, 20);
33     txtArea.setEditable(false);
34     add(txtArea);
35
36     btnOK = new JButton("確定");
37     btnOK.setBounds(20, 110, 70, 20);
38     add(btnOK);
39     btnOK.addActionListener(new ActionListener()
40     {
41         public void actionPerformed(ActionEvent e)
42         {
43             int len, area;
44             len = Integer.parseInt(txtLen.getText());
45             area = len * len;
46             txtArea.setText("面積： " + area);
47         }
48     });
49
50     btnCls = new JButton("清除");
51     btnCls.setBounds(120, 110, 70, 20);
52     add(btnCls);
```

取消註冊事件傾聽者

```
53     btnCls.addActionListener(new ActionListener()
54     {
55         public void actionPerformed(ActionEvent e)
56         {
57             txtLen.setText("0");
58             txtArea.setText("");
59         }
60     });
61
62     repaint();
63 }
64 }
65
66 public class CH09_12
67 {
68     public static void main(String[] args)
69     {
70         CAreaF areaF = new CAreaF();
71     }
72 }
```

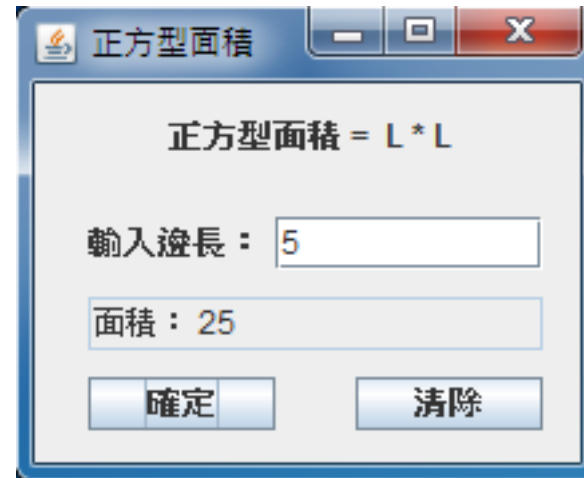
取消註冊事件傾聽者

◆ 執行結果：



正方形面積 = $L * L$

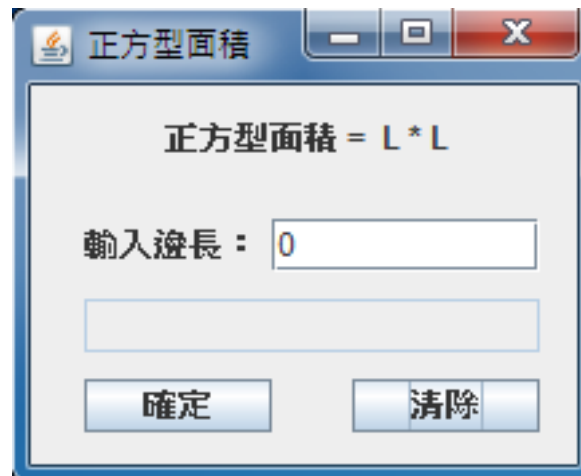
輸入邊長：



正方形面積 = $L * L$

輸入邊長：

面積：



正方形面積 = $L * L$

輸入邊長：

取消註冊事件傾聽者

◆ 說明：

- 行01：
 - ◆ 定義「套件（package）」。
- 行03：
 - ◆ 載入「javax.swing.*」套件。
- 行04：
 - ◆ 載入「java.awt.event.*」套件。
- 行06~行64：
 - ◆ 建立繼承自「JFrame」父類別的「CAreaF」子類別。
 - ◆ 行08~行09：
 - ◆ 宣告私有欄位成員。
 - ◆ 行11~行63：
 - ◆ 宣告類別「建構子」「CAreaF」。

取消註冊事件傾聽者

- ◆ 行13~行17：
 - ◆ 建立視窗物件。
- ◆ 行19~行21：
 - ◆ 建立標籤（lblTitle）物件。
- ◆ 行23~行25：
 - ◆ 建立標籤（lblLen）物件。
- ◆ 行27~行29：
 - ◆ 建立單行文字欄位（txtLen）物件。
- ◆ 行31~行34：
 - ◆ 建立單行文字欄位（txtArea）物件。
- ◆ 行36~行48：
 - ◆ 建立按鈕（btnOK）物件。
 - ◆ 行39~行48：
 - 設定事件來源物件連結事件傾聽者。
 - 此為匿名傾聽者物件。

取消註冊事件傾聽者

行41~行47：

建立actionPerformed()方法。

計算面積，並用setText()方法，將計算結果寫入txtArea中。

注意：

此為匿名傾聽者物件，所以無法取消註冊傾聽者物件。

◆ 行50~行60：

◆ 建立按鈕（btnCls）物件。

◆ 行53~行60：

設定事件來源物件連結事件傾聽者。

此為匿名傾聽者物件。

行55~行59：

建立actionPerformed()方法。

用setText()方法分別設定txtLen、txtArea的值。

取消註冊事件傾聽者

注意：

此為匿名傾聽者物件，所以無法取消註冊傾聽者物件。

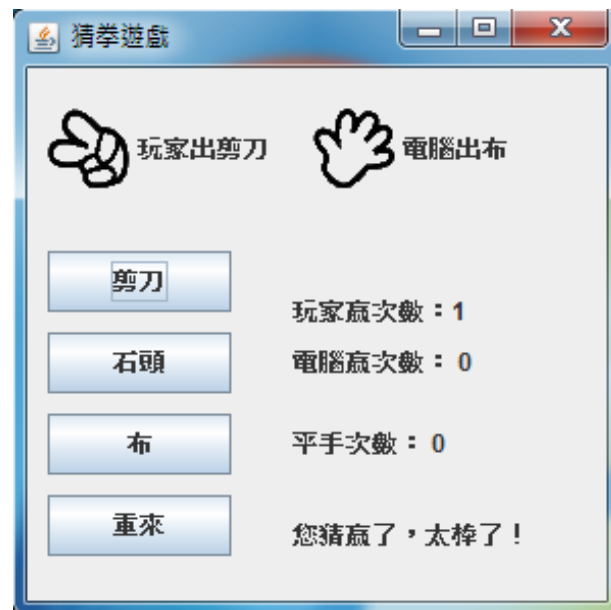
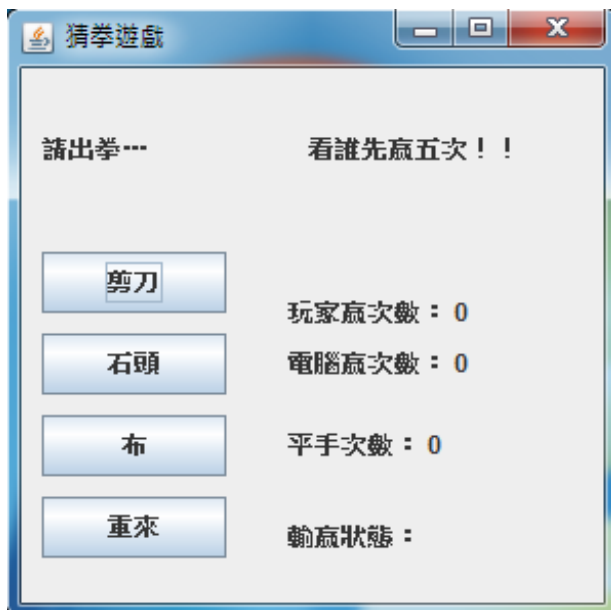
- ◆ 行62：
 - ◆ 重繪視窗元件。
- 行70：
 - ◆ 利用「CAreaF()」的「建構子」，建立areaF物件。

綜合練習

題目：猜拳遊戲

◆ 要求：

- 玩家按「剪刀」、「石頭」、「布」三個按鈕的其中一個，代表玩家出拳，電腦則隨機出拳，比對猜拳結果，以最先贏五次為最後勝利者。
- 畫面如下圖：



資料來源

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