

# COMPUTER PROGRAMMING

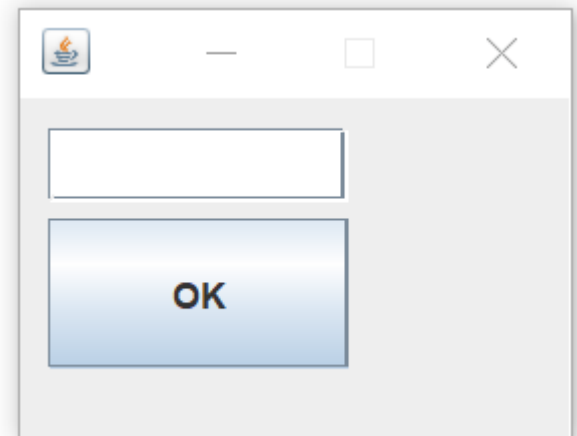
## SECTION 8 : EVENT HANDLING



# Layout แบบ null

2

```
JFrame f = new JFrame();  
Container cp = f.getContentPane();  
cp.setLayout(null);  
JTextField t1 = new JTextField();  
JButton b1 = new JButton("OK");  
t1.setBounds(10, 10, 100, 25);  
cp.add(t1);  
b1.setBounds(10, 40, 100, 50);  
cp.add(b1);  
f.setSize(100,150); // ไม่สามารถใช้ .pack() ได้  
f.setResizable(false);  
f.setVisible(true);  
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
f.setLocationRelativeTo(null); // เริ่มต้นให้เฟรมอยู่กลางจอ
```



# Event Handling

3

Event Handling คือการจัดการกับเหตุการณ์ต่างๆ (event) ที่เกิดขึ้นในโปรแกรม กับ JButton , JTextField หรืออื่นๆ ยกตัวอย่างเช่น การคลิกเมาส์ การกดแป้นพิมพ์

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

ซึ่งจำเป็นจะต้องให้คลาสสืบทอด Event interface ต่างๆ ตามความต้องการ (Interface คือคลาสที่สร้างมาเพื่อให้คลาสอื่นสืบทอดและจำเป็นต้อง Override เมธอด ที่กำหนดไว้ด้วย)

# ActionListener

4

ActionListener เป็น interface ที่จัดการกับ JButton ,JList ,JMenuItem และ JTextField เมธอดที่ต้อง Override คือ actionPerformed จะทำงานเมื่อ **กด enter หรือคลิกเมาส์**

## หลักการทำงาน

1. ให้คลาส implement interface เช่น  

```
public class Form1 implements ActionListener { ... }
```
2. Override เมธอดของ interface เช่น  

```
public void actionPerformed(ActionEvent e) { ... }
```
3. ให้ Component เรียกใช้เมธอด **addActionListener**(Obj ของคลาสที่จะใช้ Event); เพื่อเพิ่มให้ Component นั้นสามารถเกิด event ได้  

```
JButton b = new JButton("OK");  
b.addActionListener(this);
```

# ActionListener - ตัวอย่าง

5

```
class Form2 implements ActionListener {
```

```
    JTextField t1 ;
```

```
    JButton b1 ;
```

```
    public Form2(){
```

```
        JFrame f = new JFrame();
```

```
        Container cp = f.getContentPane();
```

```
        cp.setLayout(null);
```

```
        t1 = new JTextField();
```

```
        b1 = new JButton("OK");
```

```
        b1.addActionListener(this);
```

```
        t1.setBounds(40, 10, 100, 25);
```

```
        cp.add(t1);
```

```
        b1.setBounds(40, 40, 100, 50);
```

```
        cp.add(b1);
```

```
        f.setSize(60,140);
```

```
        f.setResizable(false);
```

```
        f.setVisible(true);
```

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

```
        f.setLocationRelativeTo(null);
```

```
    }
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        b1.setText(t1.getText());
```

```
    }
```

```
}
```

# ActionListener

6

กรณีมี Component หลายตัวที่อยากให้เกิด Event แล้วใช้เมธอดเดียวกันให้ทำตามนี้

1. ให้ Component เรียกใช้เมธอด `setActionCommand(ชื่อ);` เพื่อตั้งชื่อคำสั่ง เช่น

```
b1.setActionCommand("Click_B1");
```

```
b2.setActionCommand("Click_B2");
```

2. ในเมธอดของ event ต่างๆให้ใส่การเช็คเงื่อนไขว่ามาจาก Component คำสั่งชื่ออะไรจากการใช้ `getActionCommand()` เช่น

```
public void actionPerformed(ActionEvent e) {  
    if(e.getActionCommand()=="Click_B1")  
    { .... }  
    else if(e.getActionCommand()=="Click_B2")  
    { .... }  
}
```

# ActionListener - ตัวอย่าง

7

```
class Form2 implements ActionListener {
```

```
    JTextField t1 ;
```

```
    JButton b1 ;
```

```
    JButton b2 ;
```

```
    public Form2(){
```

```
        JFrame f = new JFrame();
```

```
        Container cp = f.getContentPane();
```

```
        cp.setLayout(new FlowLayout());
```

```
        t1 = new JTextField("",15);
```

```
        b1 = new JButton("OK");
```

```
        b2 = new JButton("Del");
```

```
        b1.setActionCommand("Click_B1");
```

```
        b2.setActionCommand("Click_B2");
```

```
        b1.addActionListener(this);
```

```
        b2.addActionListener(this);
```

```
        cp.add(t1);
```

```
        cp.add(b1);
```

```
        cp.add(b2);
```

```
        f.pack();
```

```
        f.setVisible(true);
```

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

```
        f.setLocationRelativeTo(null);
```

```
    }
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        if(e.getActionCommand()=="Click_B1")
```

```
            b1.setText(t1.getText());
```

```
        else if(e.getActionCommand()=="Click_B2")
```

```
            t1.setText("");
```

```
    }
```

```
}
```

# Event Interface แบบต่างๆ

8

MouseListener จัดการเกี่ยวกับ Event ของเมาส์

- mousePressed – กดปุ่ม
- mouseReleased - ปล่อยปุ่ม
- mouseClicked - กดแล้วปล่อย
- mouseEntered – เมาส์ไปอยู่ใน Component
- mouseExited – เมาส์ออกนอก Component

KeyListener จัดการกับ Event การกดแป้นพิมพ์

- keyTyped – ตรวจสอบประเภทของแป้น
- keyPressed – กดปุ่มบนแป้น
- keyReleased – ปล่อยปุ่มบนแป้น

มักใช้งานคู่กับ (e.getKeyCode() == KeyEvent.VK\_DOWN) สีแดงคือเปลี่ยนได้  
เป็นการเช็คที่กำลังกดตัวอะไร



# Event Interface แบบต่างๆ

9

TextListener จัดการกับ Event ที่มีเกี่ยวกับตัวอักษร

- textValueChanged – เมื่อมีการเปลี่ยนแปลงตัวอักษร

\*\*\* Class สามารถสืบทอด interface ได้หลายๆตัวพร้อมกัน แต่ต้อง Override เมธอดทั้งหมดของทุก interface เช่น

```
class Form1 implements ActionListener , TextListener{  
...  
    public void actionPerformed(ActionEvent e) { ... }  
    public void textValueChanged(TextEvent e) { ... }  
  
}
```

แถมยังสามารถสืบทอดคลาสแบบปกติได้ในเวลาเดียวกันอีกด้วย เช่น

```
class Form1 extends JFrame implements ActionListener , TextListener{ ... }
```

10