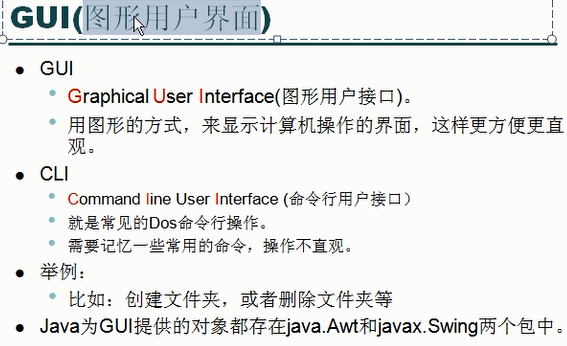
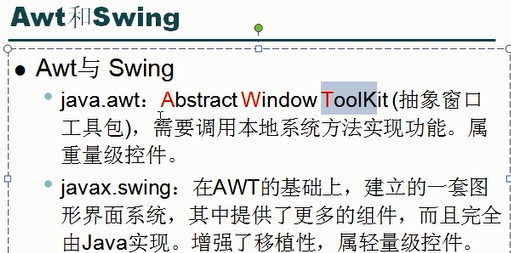
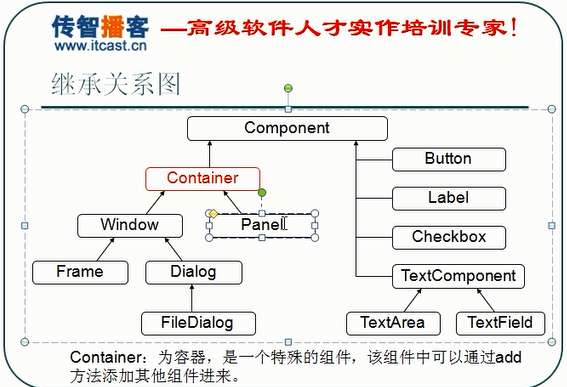
# GUI

## 01-GUI(概述)







能添加组件的组件称之为容器。



## 02-GUI(Frame演示)



import java.awt.Button;

import java.awt.FlowLayout;

import java.awt.Frame;

public class FrameDemo {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Frame f = new Frame();

// f.setLocation(400, 100);

// f.setSize(500, 400);

f.setBounds(100, 100, 500, 400);

f.setLayout(new FlowLayout());//设置流式布局

Button but = new Button("一个按钮");

f.add(but);//将按钮添加到窗体中

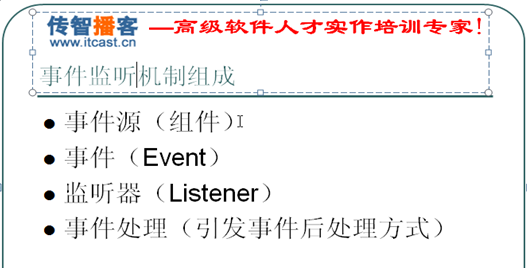
f.setVisible(true);

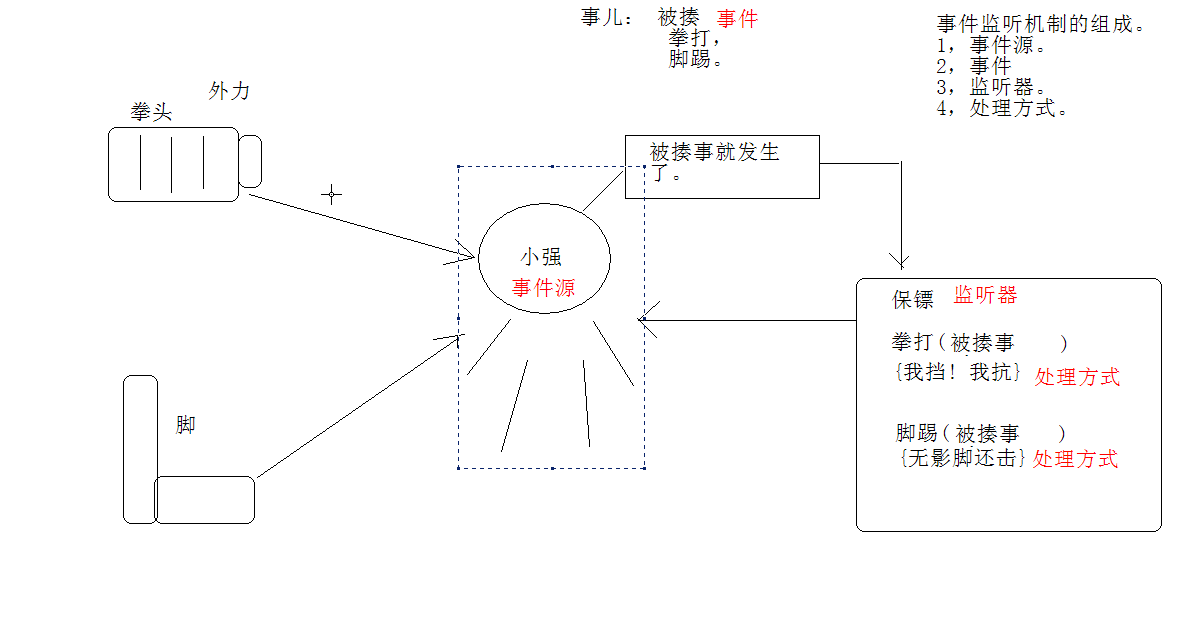
System.out.println("over");

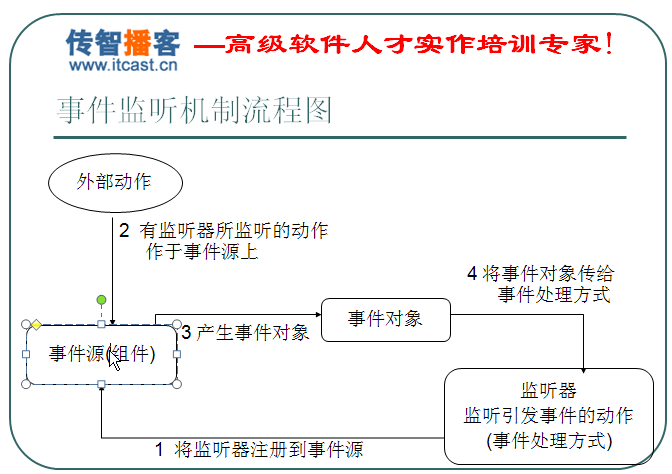
}

}

## 03-GUI(事件监听机制)

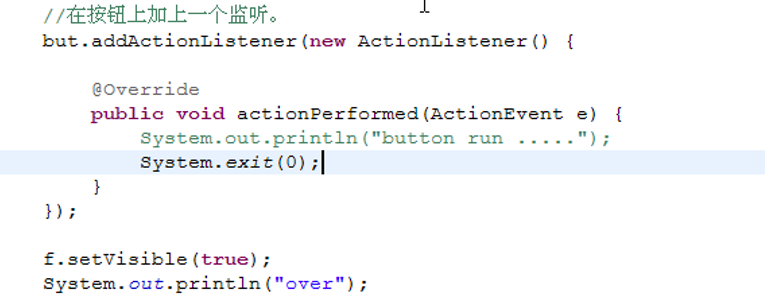




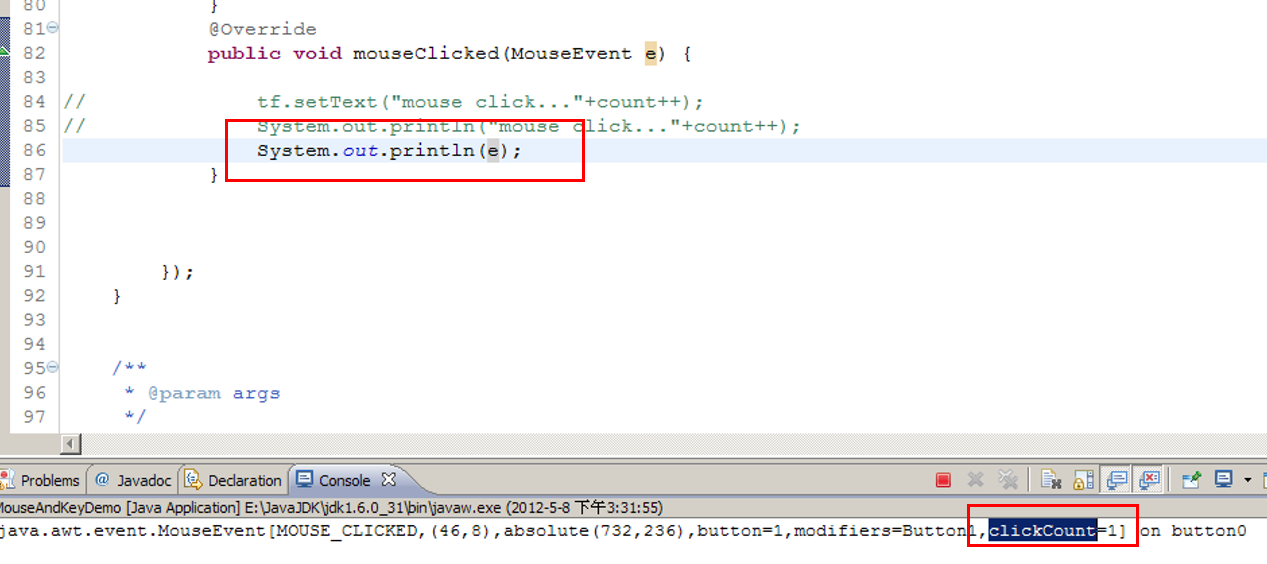


Alt+Shift+S 打开：override

## 04-GUI(ActionListener演示)



## 05-GUI(鼠标事件)



**package** GUIDemo;

**import** java.awt.Button;

**import** java.awt.FlowLayout;

**import** java.awt.Frame;

**import** java.awt.TextField;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**import** java.awt.event.WindowAdapter;

**import** java.awt.event.WindowEvent;

**public** **class** MouseAndKeyDemo {

**private** Frame f;

**private** TextField tf;

**private** Button but;

**public** MouseAndKeyDemo() {

init();

}

**private** **void** init() {

// **TODO** Auto-generated method stub

f = **new** Frame("演示鼠标和键盘");

f.setBounds(400,200,500,400);

f.setLayout(**new** FlowLayout());

tf = **new** TextField(25);

but = **new** Button("一个按钮");

f.add(tf);

f.add(but);

myEvent();

f.setVisible(**true**);

}

**private** **void** myEvent() {

// **TODO** Auto-generated method stub

f.addWindowListener(**new** WindowAdapter(){

**public** **void** windowClosing(WindowEvent e){

System.*exit*(0);

}

});

but.addActionListener(**new** ActionListener() {

@Override

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

// **TODO** Auto-generated method stub

System.*out*.println("nihao !");

}

});

//在按钮上添加一个鼠标监听。

but.addMouseListener(**new** MouseAdapter() {

**private** **int** count = 0;

@Override

**public** **void** mouseEntered(MouseEvent e) {

// **TODO** Auto-generated method stub

// System.out.println("mouse enter"+count++);

// tf.setText("mouse enter"+count++);

}

@Override

**public** **void** mouseClicked(MouseEvent e) {

// **TODO** Auto-generated method stub

// tf.setText("mouse click "+count++);

// System.out.println("mouse enter"+count++);

**if**(e.getClickCount()==2){

tf.setText("mouse double click "+count++);

}

}

});

}

/\*\*

\* **@param** args

\*/

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

// **TODO** Auto-generated method stub

**new** MouseAndKeyDemo();

}

}

## 06-GUI(键盘事件)

事件监听其实很简单：

第一步：确定事件源

第二步：确定事件和监听器，你知道你要确定什么事件吗？鼠标，那就用鼠标监视器！动作，action监视器！键盘，key监视器！

第三步：确定具体的动作，点击、释放

第四步：把内容写进去。

一般事件对应的动作都不唯一，除少数几个，监听器一般都是多方法的，一般都具备适配器。



## 07-GUI(Swing演示&装插件)

## 08-GUI(练习-列出目录内容)

