# 对话框

## 实验目的

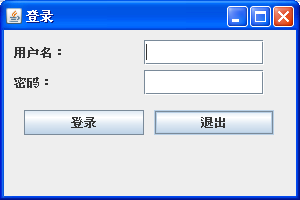
掌握Java中常用的几种模式对话框的使用方法

## 前提条件

1. 配置完成Eclipse开发环境；
2. 对java中Swing中的常用组件和事件等有一定的了解。

## 实验任务及完成标准

任务：完成一个如图所示的界面：点击登陆按钮时，检测用户名和密码是否正确（用户名：admin；密码：admin），如果正确，弹出选项对话框，选择用户的身份（学生、教师、教授），并在控制台打印用户以何种身份登录（如：您以学生身份登录系统）；如果不正确，用提示信息对话框，提示用户名或密码错误。点退出按钮时，弹出确认信息对话框，询问是否真的退出程序，如果用户选择是，则退出程序。



注：创建界面的过程为：声明组件--设置组件的位置和属性--将组件添加到指定的容器中—在窗体上显示组件。

步骤：

1. 打开eclipse。
2. 新建一个类文件。
3. 因为是要在窗体中显示所以创建一个JFrame的子类，这样就可以直接从JFrame中继承窗体所有的属性。
4. 在新建的类文件中定义一个类并命名为LoginFrame。在类中声明所需的组件。

JPanel loginPanel = **null**;

JLabel lblName = **null**;

JLabel lblPwd = **null**;

JTextField txtName=**null**;

JPasswordField txtPwd=**null**;

JButton btnLogin = **null**;

JButton btnCancle = **null**;

1. 声明好所需的组件以后就可以在构造函数中实例化该组件并且设置该组件的属性等。

在构造函数中可以通过以下步骤来使用这么声明的组件

将盛放组件的容器的布局设置为null----实例化组件----定义组件的属性和位置-----为组件添加事件----将组件添加到容器中-----显示组件

JPanel loginPanel = **null**;

JLabel lblName = **null**;

JLabel lblPwd = **null**;

JTextField txtName=**null**;

JPasswordField txtPwd=**null**;

JButton btnLogin = **null**;

JButton btnCancle = **null**;

1. 将盛放组件的容器的布局设置为null：

**super**("登录");//设置窗体的标题

loginPanel.setLayout(**null**) ;

1. 实例化各组件：

loginPanel = **new** JPanel();

//初始化标签对象

lblName=**new** JLabel("用户名：");

lblPwd=**new** JLabel("密码：");

//初始化文本框对象

txtName=**new** JTextField();

txtPwd=**new** JPasswordField();

//初始化按钮对象

btnLogin=**new** JButton("登录");

btnCancle=**new** JButton("取消");

1. 定义组件的属性和位置：

lblName.setBounds(10, 10, 120, 25);

lblPwd.setBounds(10, 40, 120, 25);

txtName.setBounds(140, 10, 120, 25);

txtPwd.setBounds(140, 40 ,120, 25);

btnLogin=**new** JButton("登录");

btnCancle=**new** JButton("取消");

1. 将组件添加到指定的容器：

//将组件添加到面板

loginPanel.add(lblName);

loginPanel.add(lblPwd);

loginPanel.add(txtName);

loginPanel.add(txtPwd);

loginPanel.add(btnLogin);

loginPanel.add(btnCancle);

1. 将容器添加到窗体上并显示组件和窗体。

**this.**getContentPane().add(loginPanel);

**this**.setSize(300, 200);

**this**.setVisible(**true**);

1. 编写登录和退出的方法实现（login(),exit()）

//登录方法的实现

**public** **void** login(){

String statue[]=**new** String[]{"学生","教师","教授"};

**if**(txtName.getText().equals("admin") && txtPwd.~~getText~~()

.equals("admin")){

**int** num=JOptionPane.*showOptionDialog*(**this**

, "请选择您的身份", "选择"

, JOptionPane.*YES\_NO\_OPTION*

, JOptionPane.*INFORMATION\_MESSAGE*

, **null**, statue, "2");

**if**(num >= 0){

System.*out*.println("您以" + statue[num]

+ "身份登录系统");

}

}**else**{

JOptionPane.*showMessageDialog*(**this**

, "您输入的用户名和密码不正确！"

,"提示",JOptionPane.*ERROR\_MESSAGE*);

}

}

//退出方法的实现

**public** **void** exit(){

**if**(JOptionPane.*showConfirmDialog*(**this**

, "您真的要退出吗？"

,"退出",JOptionPane.*YES\_NO\_OPTION*

,JOptionPane.*QUESTION\_MESSAGE*

)==JOptionPane.*YES\_OPTION*){

System.*exit*(0);

}

}

1. 在主函数中生成类的实例。

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

LoginFrame loginFrame=**new** LoginFrame();

}

1. 编写监听器类BtnListener，并实现ActionListener接口。要求编写含参数的构造方法，并且参数的类型为LoginFrame，定义LoginFrame类型的引用的属性，在构造方法中初始化。

LoginFrame loginFrame;

**public** BtnListener(LoginFrame loginFrame){

**this**.loginFrame=loginFrame;

}

1. 实现actionPerformed方法，判断用户点击的是登录按钮还是取消按钮，从而调用窗体不同的方法。

@Override

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

**if**(e.getActionCommand().equals("登录")){

loginFrame.login();

}**else** **if**(e.getActionCommand().equals("取消")){

loginFrame.exit();

}

}

1. 在LoginFrame页面，为两个按钮注册动作事件：

BtnListener listener=**new** BtnListener(**this**);

btnLogin.addActionListener(listener);

btnCancle.addActionListener(listener);