## 实验8：图形用户界面的使用

### 1.实验目的

（1）掌握Java的基本图形用户界面组件的使用方法。

（2）掌握Java图形用户界面事件处理方法。

### 2.实验内容

（1）在JFrame窗口中，加入一个内容面板，在该面板上加入一个标签、一个按钮、一个文本框，使用FlowLayout布局。设置标签、按钮、文本框的前景色、背景色、字体等属性。

1. 源代码：

package shiyan\_8;

import java.util.\*;

import java.lang.\*;

import javax.swing.\*; //引入swing包

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class test1 {

public static void main(String[] args) {

JFrame frame = new JFrame();

JLabel jt = new JLabel(" Try your best ?"); // 标签

Font f = new Font("楷体", Font.BOLD, 26); //加粗，楷体

jt.setFont(f);

jt.setForeground(Color.RED); // 设置标签上的文字的颜色，默认颜色是黑色

jt.setOpaque(true); // 此句是重点，设置背景颜色必须先将它设置为不透明的，因为默认是透明的。。。

jt.setBackground(Color.YELLOW); // 设置标签的背景颜色

JButton yes = new JButton("yes!");// 创建按钮

Font butFon = new Font("宋体", Font.BOLD, 16); //宋体，正常

yes.setFont(butFon);

yes.setForeground(Color.BLACK);

JTextField txt = new JTextField(" 那就一定要加油哦！");//文本框

Font txt\_fon = new Font("楷体", Font.PLAIN, 12); //正常，楷体

txt.setFont(txt\_fon);

txt.setForeground(Color.BLUE);

FlowLayout flow = new FlowLayout();// 创建流式布局

frame.setLayout(flow);

frame.add(jt);

frame.add(yes);

frame.add(txt);

frame.setBounds(300, 300, 200,150 );

frame.setVisible(true);

// 添加按钮的点击事件监听器

yes.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// 获取到的事件源就是按钮本身

System.exit(0);

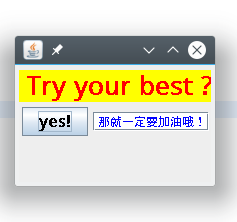
}

});

}

}

1. 运行截图：



1. 使用Swing组件做一个登录界面。要求有用户名、密码文本框及标签，两个按钮（登录和退出），并能够响应鼠标单击按钮事件，即当单击“登录”按钮时显示“登录成功”，单击“退出”按钮时退出程序。

（提示：可调用JFrame组件的dispose()方法退出程序。）

1. 源代码：

package shiyan\_8;

import java.util.\*;

import java.lang.\*;

import java.awt.BorderLayout; //BorderLayout(边界布局)

import java.awt.GridLayout;//GridLayout(网格布局)

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

import javax.swing.SwingConstants;

/\*extends 继承　，\*/

public class test2 extends JFrame implements ActionListener {

JButton login = new JButton("login in");

JButton exit = new JButton("login out");

JLabel name = new JLabel("username：");

JLabel password = new JLabel("password：");

/\* 文本框的长度为10列，即当用户输入的字符超过10个时数据就会超出可视范围，但是文本框不会限制用户只能输入10个字符 \*/

JTextField JName = new JTextField(13); // 明文账号输入

JPasswordField JPassword = new JPasswordField(13); // 非明文密码输入

public test2() {

JPanel jp = new JPanel();

jp.setLayout(new GridLayout(3, 2)); // 3行2列的面板jp（网格布局）

name.setHorizontalAlignment(SwingConstants.RIGHT); // 设置该组件的对齐方式为向右对齐

password.setHorizontalAlignment(SwingConstants.RIGHT);

jp.add(name); // 将内容加到面板jp上

jp.add(JName);

jp.add(password);

jp.add(JPassword);

jp.add(login);

jp.add(exit);

login.addActionListener(this); // login in 增加事件监听

exit.addActionListener(this); // login out 增加事件监听

this.add(jp, BorderLayout.CENTER); // 将整块面板定义在屏幕中间

this.setTitle("login window");

this.setLocation(500, 300); // 设置初始位置

this.pack(); // 表示随着面板自动调整大小

this.setVisible(true);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public void actionPerformed(ActionEvent e) // => try catch 语句

{

if (e.getSource() == exit) {

this.dispose();

} else {

if(JName.getText().equals("Tattoo") && String.valueOf(JPassword.getPassword()).equals("Tattoo")) {

JOptionPane.showMessageDialog(null, " Success! Welcome ");

// 显示信息提示框

this.dispose();

} else {

JOptionPane.showMessageDialog(null, "failed! Please try again...");

// 显示信息提示框

JName.setText("");

JPassword.setText("");

}

}

}

public static void main(String[] args) {

JFrame.setDefaultLookAndFeelDecorated(false); // 不禁用本地外观

new test2();

}

}

1. 运行截图：



1. 实验总结：

这次实验主要是java.awt(与系统有关)包和javax.swing包(基于JVM,在任何平台运行效果都一样)中一些基础的组件的使用，主要是多熟悉，多练习，就能很好的掌握了．希望自己以后多多练习．