**Swing**

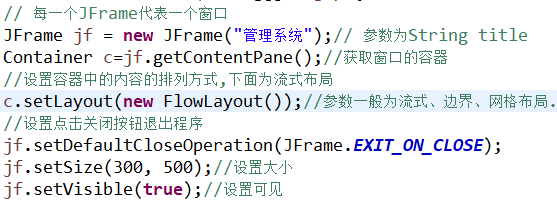
导包：**import javax.swing.\*；import java.awt.\*；**

一、Swing

**JFrame**代表一个窗口。**Container 、getContentPane、setLayout、FlowLayout、setDefaultCloseOperation、setSize、setVisible。**

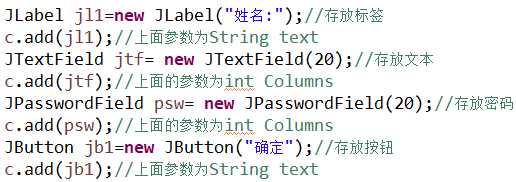
**注意这些Swing主键不一定要写在main方法中，最好写在一个XxxView的类中，然后在main程序中创建该XxxView的对象。**

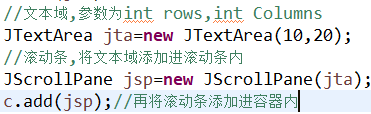
注意设置JFrame的大小、可见性、关闭的程序一般放在整个代码的最下面，否则可能出错。



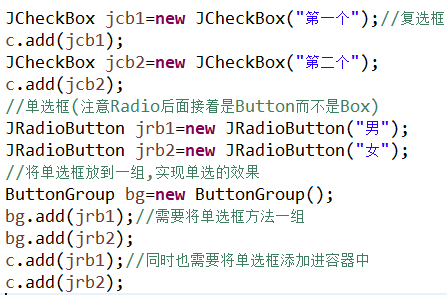
**JLabel**标签，注意这个Label单词自己经常写错，写成Lable。

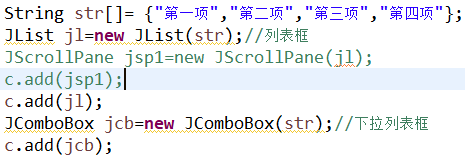
**JTextField、JPasswordField、JButton、JTextArea、JScrollPane**

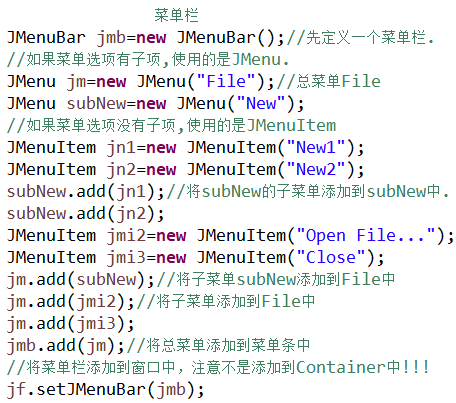




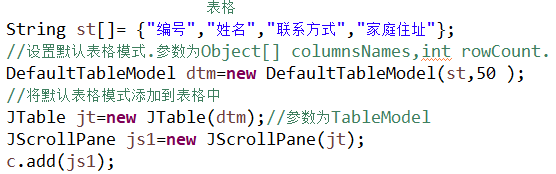
**JCheckBox、JRadioButton、ButtonGroup、JList、JComboBox(下拉列表框)、JMenuBar、JMenu、JMenuItem、setMenuBar**

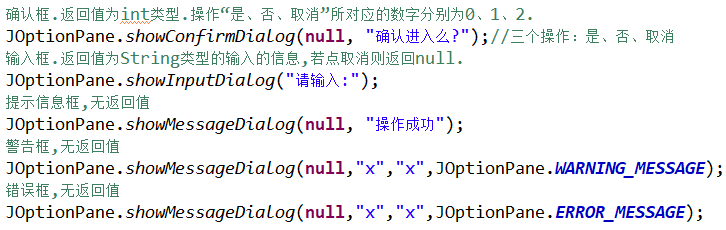






**DefaultTableModel、JTable、JOptionPane.showConfirmDialog、JOptionPane.showInputDialog、JOptionPane.showMessageDialog**

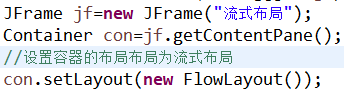






二、布局

1.流式布局：从左到右依次排列。

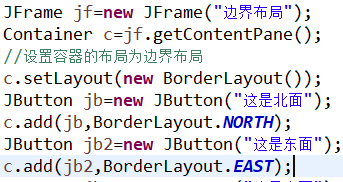


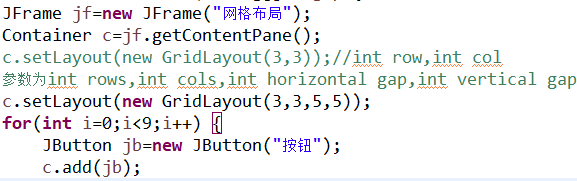
2.边界布局：将整个窗口分为东南西北中五块区域，其中，中间的区域如果不指定,会被预留出来。

3.网格布局。

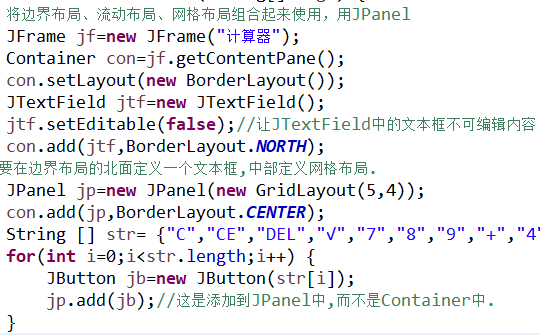
4.将这些布局组合起来使用，用**JPanel**画布。先定义一个画布JPanel，

然后在JPanel中再重新定义一个布局，定义完之后，再将JPanel以某种布局添加到容器Container中。注意!!!在画布中定义的新布局，在布局中添加内容时,最后要将添加的内容add在JPanel中，而不是Container中。



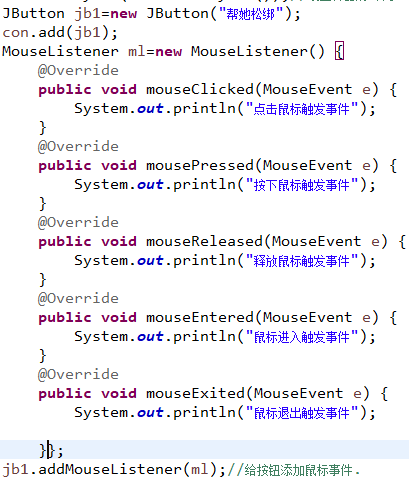


[**JText**Component](eclipse-javadoc:%E2%98%82=LayoutTest/E:%5C/JavaJDK%5C/lib%5C/jrt-fs.jar%60java.desktop%3Cjavax.swing.text(JTextComponent.class%E2%98%83JTextComponent)**.setEditable(boolean b)**：将JText类型的组件设置编辑模式，如参数为false时，表不可编辑。



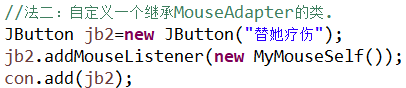
三、鼠标事件

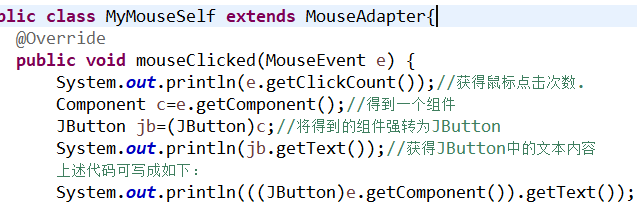
MouseListener是一个接口，MouseAdapter是一个抽象类，其实现了MouseListener接口。实现鼠标事件有两种方法，**法一**：定义一个MouseListener匿名内部类，**重写该类的全部方法**。然后再将其添加到要触发鼠标事件的如按钮中去。法二：自定义一个类，然后继承MouseAdapter，并**重写需要使用的方法**。法三：定义一个MouseAdapter()匿名内部类，并重写需要使用的方法。



MouseEvent**.getClickCount()**：无参数，返回值为int类型，返回的是点击鼠标事件的次数。若一直点击则次数会一直增加，若点一次停一次这样次数就一直是1。

ComponentEvent**.getComponent()**：无参数，返回值为Component。ComponentEvent可以翻译成组件事件，如MouseEvent是鼠标事件，可以归为ComponentEvent中。getComponent()方法是获得组件，如JButton、JTextField等等都是一个组件。



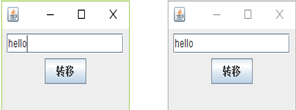


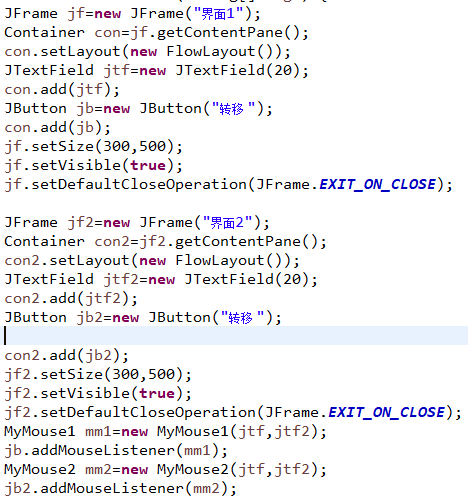
四、键盘事件

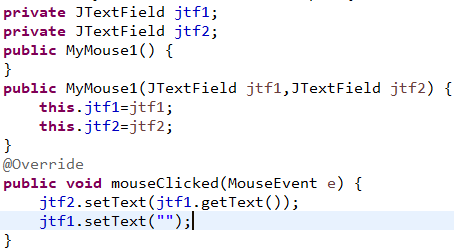
键盘事件同鼠标事件类似。KeyListener 是一个接口，KeyAdapter是一个实现了KeyListener接口的抽象类。**法一：**定义一个KeyListener匿名内部类，重写全部方法，再将该键盘事件添加到组件中。**法二：**自定义一个类并继承KeyAdapter，重写需要的方法。

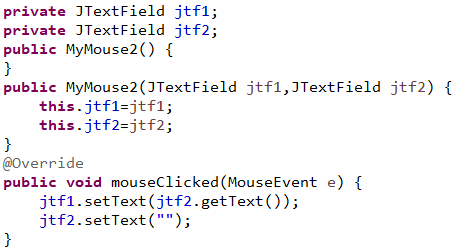
KeyEvent**.getKeyChar()**：返回值为char类型，以字符的形式单个单个的返回键盘输入的内容。

**作业**：实现两个JFrame中的内容相互转移。









**五、Swing内容补充：ActionListener事件**

Window**.setLocation(int x，int y)**：给窗口设置位置，无返回值，参数x为水平方向，y为垂直方向。

ActionListener是一个接口，只有一个actionPerformed()方法。

ActionEvent**.getActionCommand()**：返回值为String类型，以String类型返回标识此事件的命令。如将按钮JButton添加了ActionListener事件，在重写的actionPerformed()方法中，e.getActionCommand()得到的就是对应按钮上的文字。

JTable中没有增加、删除表格行、列等操作，但是DefaultTableModel中有。

JTable**.getModel()**：返回值为TableModel类型。返回的TableModel上提供了JTable中的数据。即对Table中的数据进行增加、删除等操作用要使用TableModel模式。可将该模式强制转换成DefaultTableModel模式。这是JTable的方法。

JTable**.getSelectedRow()**：返回值为int类型，返回的是第一个选定行的索引，若没有选中行，则返回的是-1。

JTable**.getSelectedRows()：**返回值为int[]类型，返回的是所有选中行的索引。若没有选中的行，则返回的是一个空数组。

JTable**.getTableHeader()**：返回值为JTableHeader，获得表格的表头。

JTable**.columnAtPoint(Point point)**：返回值为int类型。当点击列时得到该列的索引位置。**列的索引是[0，getColumnCount()-1]**，当索引不在该范围，则返回的是-1。

JTable**.getColumnName(int columnIndex)**：返回值为String类型，返回的是索引号所对应的列的名称。

DefaultTableModel**.addRow(object[] rowdata)**：给表格的行添加数据，无返回值。

DefaultTableModel**.removeRow(int index)：**删除指定索引位置上的行。

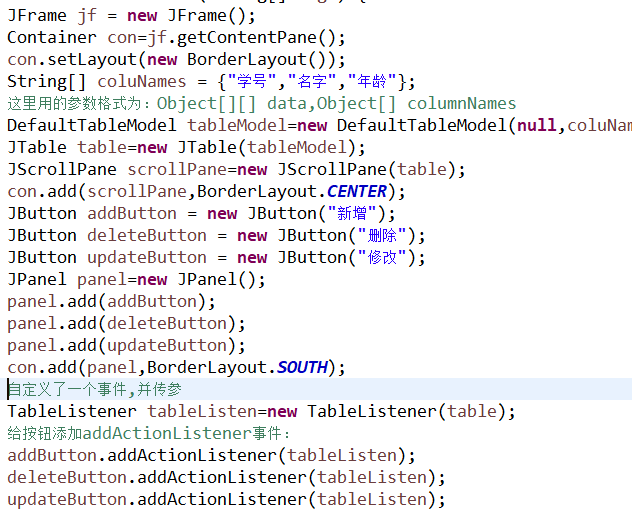
DefaultTableModel**.getValueAt(int row，int colum)**：返回值为object，得到表格中对应行与列所在的单元格的内容，即单元格的值。一般强转为String类型、int等等类型的数据。

DefaultTableModel**.setValueAt(object value，int row，int colum)：**无返回值，给表格中对应行与列所在的单元格的设置值，第一个参数为给单元格设置的值。

DefaultTableModel**.setRowCount(int num)**：给表格设置行，如果设置的行数比原来的表格行数多则添加行，若少则删除行，为0是清空表格内容。。

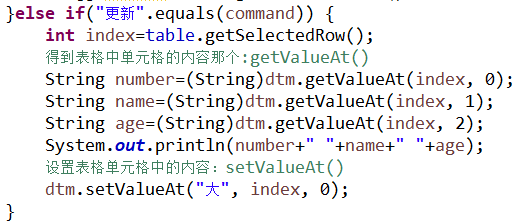
MouseEvent**.getPoint()**：返回值为Point，得到的是点击的位置。

实例：







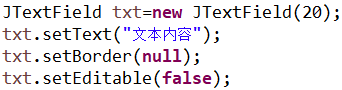


总结：对表格中内容的修改、增加、删除行都是在DefaultTableModel模式下进行的。得到表格的模式、表头、列行名都是在JTable下进行。

**六、内容补充**：

1. [JTextField](eclipse-javadoc:%E2%98%82=PersonnelManagementSystem/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing(JTextField.class%E2%98%83JTextField)**.setHorizontalAlignment(int alignment)**：可设置JTextField中的文本水平对齐。参数可为：JTextField***.LEFT***、JTextField***. CENTER*** 、JTextField***.RIGHT*** 、JTextField***.LEADING*** 、JTextField***.TRAILING***。

2. [JComponent](eclipse-javadoc:%E2%98%82=testPro/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing(JComponent.class%E2%98%83JComponent)**.setBorder(**[**Border**](eclipse-javadoc:%E2%98%82=testPro/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing(JComponent.class%E2%98%83JComponent~setBorder~Ljavax.swing.border.Border;%E2%98%82javax.swing.border.Border) **border)**：可设置组件的边界，当参数为null时边界没有边界。[JComponent](eclipse-javadoc:%E2%98%82=testPro/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing(JComponent.class%E2%98%83JComponent)**.setEditable(boolean b)**：可设置组件是否可编辑。一般可将这两者结合使用，单独设置组件的边界为null时，该组件的背景会编程纯白色。将JTextField设置成如下格式，在页面窗口中显示结果右边所示：



3. [JTextComponent](eclipse-javadoc:%E2%98%82=PersonnelManagementSystem/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing.text(JTextComponent.class%E2%98%83JTextComponent)**.setText(**[**String**](eclipse-javadoc:%E2%98%82=PersonnelManagementSystem/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing.text(JTextComponent.class%E2%98%83JTextComponent~setText~Ljava.lang.String;%E2%98%82java.lang.String) **t)**：给Text类型的组件设置文本内容，如JTextField、JTextArea、JButton、JPasswordField等组件。[JTextComponent](eclipse-javadoc:%E2%98%82=PersonnelManagementSystem/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing.text(JTextComponent.class%E2%98%83JTextComponent)**.getText()**：获得Text类型的组件的文本内容。

4. JComBox**.setSelectedItem(Object anobject)**：设置下拉列表框中选中的内容，当为null则什么也不选中。JComBox**.getSelectedItem()**：获得下拉列表框中选中的内容。JComBox.getSelectedIndex()：获得下拉列表框中选中内容的索引。

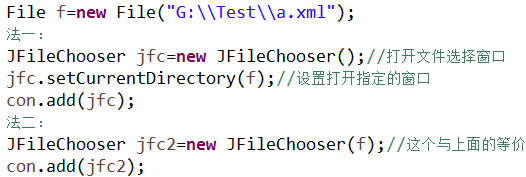
5. [System](eclipse-javadoc:%E2%98%82=testPro/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjava.lang(System.class%E2%98%83System).exit(int status)：终止正在运行的java虚拟机。

6. WindowListener是一个接口、 WindowAdapter是一个抽象类，用来监听窗口的，用来监听窗口的打开关闭等情况。

**七、文件选择的操作**：

1.JFileChooser对象：打开文件选择窗口，不加参数是打开系统默认的窗口。参数可以是File类型的，表示打开的窗口是在File目录下。

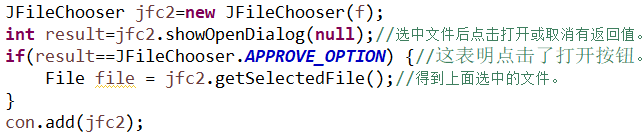
2. JFileChooser.**setCurrentDirectory(**[**File**](eclipse-javadoc:%E2%98%82=testPro/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing(JFileChooser.class%E2%98%83JFileChooser~setCurrentDirectory~Ljava.io.File;%E2%98%82java.io.File) **dir)**：设置JFileChooser打开的窗口，参数是File类型的目录。



3. JFileChooser**. showOpenDialog(null)**：选中指定文件后点击**打开**按钮或取消按钮。该方法的返回值为int类型的，返回的结果一般为：JFileChooser***.CANCEL\_OPTION*** 、JFileChooser***. APPROVE\_OPTION*** 、JFileChooser***.ERROR\_OPTION***。分别代表点击了取消按钮、点击了同意打开、打开文件出错。

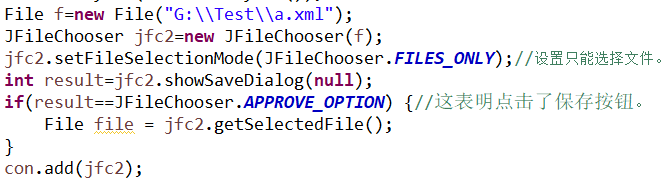
4. [JFileChooser](eclipse-javadoc:%E2%98%82=PersonnelManagementSystem/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing(JFileChooser.class%E2%98%83JFileChooser)**.getSelectedFile()**：返回的是选中的文件。返回值为File类型。[JFileChooser](eclipse-javadoc:%E2%98%82=PersonnelManagementSystem/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing(JFileChooser.class%E2%98%83JFileChooser)**.getSelectedFiles()**返回的是一个File[]，返回的是选中的多个文件。





5. [JFileChooser](eclipse-javadoc:%E2%98%82=PersonnelManagementSystem/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing(JFileChooser.class%E2%98%83JFileChooser)**.setFileSelectionMode(int mode)**：设置用户可以选择的文件类型，如文件夹、文件。参数一般为： JFileChooser***.FILES\_ONLY***、JFileChooser***.DIRECTORIES\_ONLY***、JFileChooser***.FILES\_AND\_DIRECTORIES***。

6. [JFileChooser](eclipse-javadoc:%E2%98%82=PersonnelManagementSystem/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing(JFileChooser.class%E2%98%83JFileChooser)**.showSaveDialog(null)**：选中指定文件后点击**保存**按钮或取消按钮。用法和JFileChooser.showOpenDialog(null)的用法一样。



7. [JFileChooser](eclipse-javadoc:%E2%98%82=PersonnelManagementSystem/E:%5C/JavaRunEnviroment%5C/lib%5C/rt.jar%3Cjavax.swing(JFileChooser.class%E2%98%83JFileChooser)**.**[**setMultiSelectionEnabled**](https://blog.csdn.net/liang5630/article/details/25651491)**(boolean b)**：设置是否允许选择多个文件，默认值是false。