**Java Swing** javax.swing.\*; import java.awt.\*;

GUI（Graphic User Interface）为程序提供图形界面，它最初的设计目的是构建一个通用的GUI,使其能在所有平台上运行。在Java1.0中基础类AWT（Abstract Windowing Toolkit）并没有达到这个要求，于是在Java2.0之后的版本出现了Swing，它是AWT组件的增强组件，但它并不能完全代替AWT组件，这两种组件需要同时出现在一个图形用户界面中。

AWT(Abstract Windowing Toolkit)，中文译为抽象窗口工具包，是Java提供的用来建立和设置Java的图形用户界面的基本工具。AWT由Java中的java.awt包提供，里面包含了许多可用来建立与平台无关的图形用户界面(GUI)的类，这些类又被称为组件(components)。

Swing是一个用于开发Java应用程序用户界面的开发工具包。它以抽象窗口工具包（AWT）为基础使跨平台应用程序可以使用任何可插拔的外观风格。Swing开发人员只用很少的代码就可以利用Swing丰富、灵活的功能和模块化组件来创建优雅的用户界面。

JComponent

JFrame

JTeatArea

JDialog

Dialog

Frame

JPanel

JTable

JTextField

JTree

JButton

Window

Container

Component

AWT是重量级组件，操作系统不同表现不同。Swing组件不受影响，以J开头的组件都为swing组件

**窗口**

窗口是GUI编程的基础，Java提供的JFrame类的实例是一个底层容器，即通常所说的窗口。其他组件必须被添加到底层容器中，以便借助这个底层容器和操作系统进行信息交互。

Applet窗口：Applet类管理这个窗口，当应用程序程序启动时，由系统创建和处理；

框架窗口(JFrame)：这是通常意义上的窗口，它支持窗口周边的框架、标题栏，以及最小化、最大化和关闭按钮；也是container的间接子类，但需要一个窗口时，可使用JFrame或子类创建一个对象

一种无边框窗口(JWindow)：没有标题栏，没有框架，只是一个空的矩形。

Frame类的主要构造方法：

JFrame()：创建无标题的窗口对象；

JFrame(String s)：创建一个标题名是字符串s的窗口对象。

JFrame类的其他常用方法：

setBounds(int x,int y,int width,int height)：参数x,y指定窗口出现在

屏幕的位置；参数width,height指定窗口的宽度和高度。单位是像素。

setSize(int width,int height)：设置窗口的大小，参数width和height指定窗口的宽度和高度，单位是像素。

setLocation(int x,int y)：设置窗口位置，默认位置是（0，0）

setVisible(boolean b):设置窗口是否可见，窗口默认不可见，true 可见，false 不可见

setReaizable(boolean b):设置窗口是否可调整大小，窗口默认可调整dispose（）：撤销当前窗口，并释放当前窗口所需要的资源setExtendedState(int state)：设置窗口的拓展状态

MAXIMIZED\_HORIZ（水平方向最大化）

MAXIMIZED\_VERT（垂直方向最大化）

MAXIMIZED\_BOTH（水平、垂直方向最大化）

setDefaultCloseOperation(int operation)：该方法用来设置单击窗体右上角的关闭图标后，程序会做出怎么样的处理，其中参数取下列int型的static的常量，程序根据参数operation取值作出不同处理

DO\_NOTHING\_ON\_CLOSE（什么也不做）

HIDE\_ON\_CLOSE（隐藏当前窗口）

DISPOSE\_ON\_CLOSE（隐藏当前窗口，并释放窗体占有的其他资源）

EXIT\_ON\_CLOSE（结束窗口所在的应用程序）

setBackground(Color c)：以参数 c设置窗口的背景颜色。

setForeground(Color c)：设置前景色为 c

vid setTitle(String title)：设置窗口中的标题

|  |  |
| --- | --- |
| **字段摘要** | **颜色** |
| static Color black/BLACK | 黑色 |
| static Color blue/BLUE | 蓝色 |
| static Color cyan/CYAN | 青色 |
| static Color darkGray/DARK\_GRAY | 深灰色 |
| static Color gray/GRAY | 灰色 |
| static Color green/GREEN | 绿色 |
| static Color lightGray/LIGHTGRAY | 浅灰色 |
| static Color magenta/MAGENTA | 洋红色 |
| static Color orange/ORANGE | 桔黄色 |
| static Color pink/PINK | 粉红色 |
| static Color red/RED | 红色 |
| static Color white/WHITE | 白色 |
| static Color yellow/YELLOW | 黄色 |

**菜单条、菜单、菜单项**

1、菜单条（JComponent类的子类JMenubar）

Frame类中将菜单条放置到窗口中的方法：setMenuBar(MenuBar bar);该方法将菜单条添加到窗口的顶端。注意：只能向窗口添加一个菜单条。

1. 菜单（JComponent类的子类JMenu）

Menu()：建立一个空标题的菜单。

Menu(String s)：建立一个指定标题s的菜单。

public void add(MenuItem item)：向菜单添加菜单项item。

public void add(String s)：向菜单增加指定的选项。

public MenuItem getItem(int n)：得到指定索引处的菜单项。

public int getItemCount()：得到菜单选项数目。

public void insert(MenuItem item,int n)：在菜单的指定位置插入菜单选项。

public void insert(String s,int n)：在菜单指定位置插入菜单选项。

public void remove(int n)：删除菜单指定位置的菜单选项。

public void removeAll()：删除菜单的所有选项。

1. 菜单项（JComponent类的子类JMenuItem）

MenuItem()：构造无标题菜单项。

MenuItem(String s)：构造有标题菜单项。

public void setEnabled(boolean b)：设置当前菜单项是否可被

选择。

public String getlabel()：得到菜单选项的名字。

public void addActionListener(ActionListener):向菜单项增加监视器，从菜单项接收行动事件（单击菜单项）。

4、嵌入子菜单

JMenu是JComponent类的子类，因此菜单本身也是一个菜单项，当把一个菜单看作菜单项添加到某个菜单中时，称这样的菜单为子菜单

5、菜单上的图标

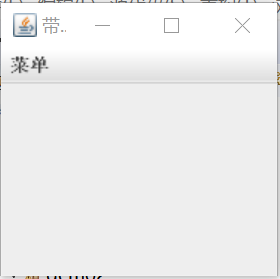
为了使一个菜单上有一个图标，可以用图标类Icon声明一个图标，使用ImageIcon类创建一个图标

Icon icon = new ImageIcon（“a.gif”);

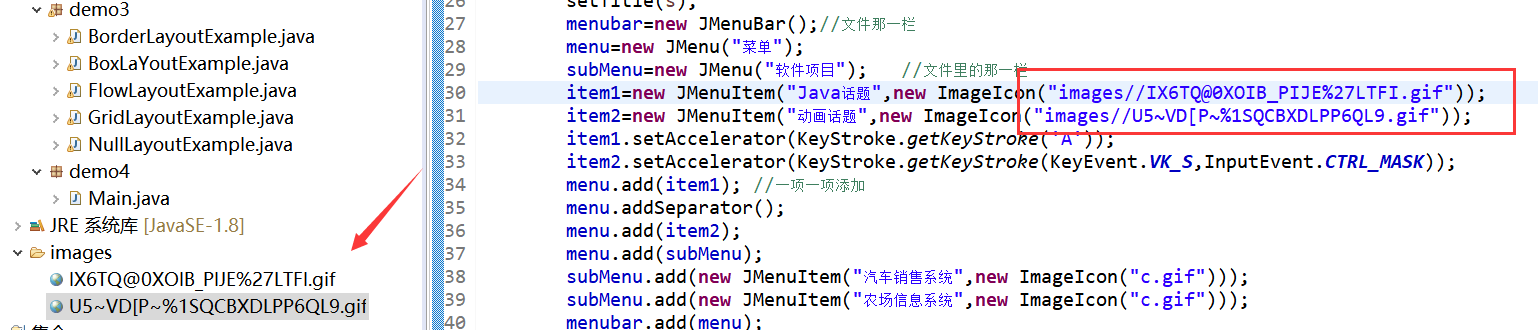
然后菜单项调用setIcon（Icon icon）方法将图标设置为icon

menu1.addSeparator(); //分割线





**添加图片要添加文件夹 再用文件夹的地址**



**常用组件**

常用组件都是JComponent的子类

JTextField:文本框：允许用户输入单行文本

JTextArea:文本区：允许用户输入多行文本

JButton：按钮：单击 按钮

JLabel：标签：为用户提供提示信息

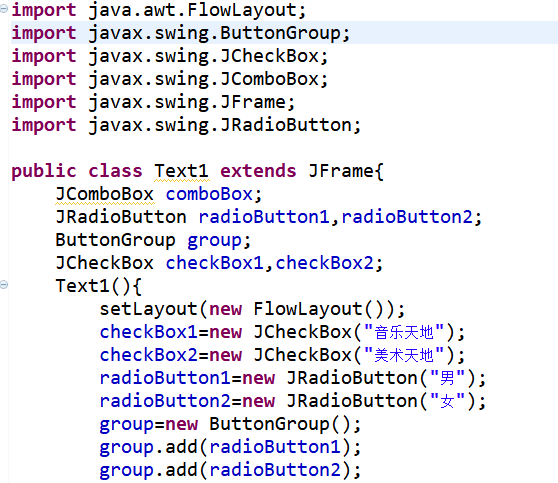
JCheckBox：复选框：为用户提供多项选择，有选中与未选中两种状态

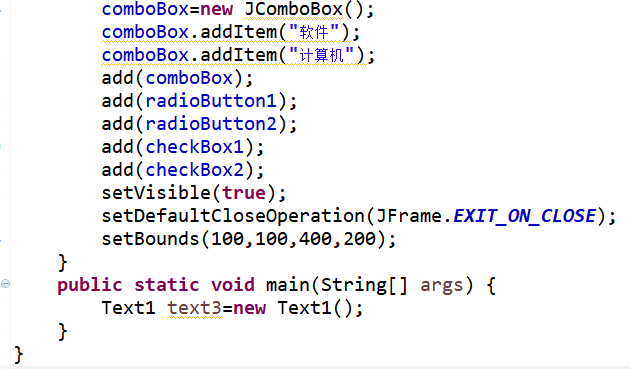
JRadioButton：单选按钮：为用户提供单项选择

JComboBox：下拉列表：单项选择

JPasswordField：密码框，默认显字符是\*，可通过setEchoChar

(char c)重新设置显字符





**常用容器**

Jcomponent是Container的子类，其创建的组件也是容器，容器经常用来添加组件，Jframe是底层容器，本届提到的容器被习惯的称作中间容器，中间容器必须添加到底层容器才能发挥作用

1. JPanel面板：默认是FlowLayout布局，可以向面板中添加组件，再把面板添加到其他容器中
2. JScrollPane：滚动窗格，可以将文本区放在滚动窗格中

JScrollPane jScrollPane=new JScrollPane(new TextArea())

1. JSplitPane：拆分窗格，可分为左右或上下两部分,有两个构造方法：

JSplitPane(int a,Component b,Component c)

//参数a决定时水平还是垂直：HORIZONTAL\_SPLIT，VERTICAL\_SPLIT

如 JSplitPane jSplitPane=new JSplitPane(JSplitPane.VERTICAL\_SPLIT,new JButton(),new JButton())

JSplitPane(int a,boolean d,Component b,Component c)

//参数d是决定拆分线移动时，组件是否连续变化，true是连续

1. JLayeredPane：分层窗格

add（Jcomponent com，int layer）;添加组件com，并指定com所在的层

setLayer(Component c，int layer)可以设置组件c所在的层

getLayer(Component c)可以获取组件c所在的层数

**常用布局**

1、FlowLayout布局（流式布局） 左到右 上到下

1. 创建布局对象 FlowLayout flow = new FlowLayout（），其对

象调用相应的方法可以重新设置布局的对齐方式

（二）容器con使用布局对象 con.setLayout（flow）

（三）Con可以使用Container类提供的add方法将组建顺序的添加到容器中

2.BorderLayout布局（边界布局） 东南西北中

（一）其布局是window型容器的默认布局

（二）使用BorderLayout布局的容器con，可以使用add方法将一个组件b添加到中心区域

Con.add（b，BorderLayout.CENTER）或

Con.add（BorderLayout.CENTER，b）

3、CardLayout布局（卡片布局）

1. 创建布局对象
2. 容器con使用布局对象  
   （三）容器调用add（String s ，Component b）将组件b添加到容器，并给出了现实该组件代号s
3. 布局对象card用CradLayout类提供的show（）方法，显示容器con中组件代号为s的组件 card.show（con，s）；

4.GridLayout布局（网格布局）

（一）创建布局对象，行m列n GridLayout gird = new GridLayout（10，8）

（二）容器调用add（Component c）将组件c加入容器中

5.null布局（空布局）

可以准确定位组件在容器的位置和大小，组件调用setBounds（int a，int b，int width，int height）方法可以设置本身的大小和在容器中的位置

# **JTabbedPane（选项卡面板）**

IMG_256

常用构造方法:

tabPlacement: 选项卡标题的位置, 值为 JTabbedPane.TOP、JTabbedPane.BOTTOM、 JTabbedPane.LEFT 或 JTabbedPane.RIGHT, 默认为 TOP

tabLayoutPolicy: 选项卡位置不能放入所有的选项卡时，放置选项卡的策略，值为 JTabbedPane.WRAP\_TAB\_LAYOUT 或 JTabbedPane.SCROLL\_TAB\_LAYOUT

JTabbedPane()

JTabbedPane(int tabPlacement)

JTabbedPane(int tabPlacement, int tabLayoutPolicy)

创建选项卡:

title: 选项卡的标题

icon: 选项卡的图标

tip: 当鼠标移到该选项卡时，悬浮提示的文本

component: 选中该选项卡时显示的内容组件

void addTab(String title, Component component)

void addTab(String title, Icon icon, Component component)

void addTab(String title, Icon icon, Component component, String tip)

在指定索引位置插入一个选项卡

void insertTab(String title, Icon icon, Component component, String tip, int index

移除选项卡:

移除指定位置的选项（下面两个方法完全一致）

void removeTabAt(int index)

void remove(int index)

移除指定内容控件的选项卡

void remove(Component component)

移除所有选项卡

void removeAll()

选项卡相关操作:

设置当前选中的选项卡

void setSelectedIndex(int index)

获取当前选中的选项卡索引

int getSelectedIndex()

获取当前选中的选项卡对应的内容组件

Component getSelectedComponent()

设置 index 位置的选项卡的标题

void setTitleAt(int index, String title)

设置 index 位置的选项卡的图标

void setIconAt(int index, Icon icon)

设置 index 位置的选项卡是否可用

void setEnabledAt(int index, boolean enabled)

将 index 位置的内容组件设置为 component

void setComponentAt(int index, Component component)

获取选项卡的数量

int getTabCount()

自定义选项卡标题位置的组件, 这里的 titleComponent 将放到选项卡标题位置。

应用场景: 例如, 在选项卡标题位置添加一个关闭按钮, 点击按钮移除选项卡。

void setTabComponentAt(int index, Component titleComponent)

状态监听器:

添加选项卡选中状态改变的监听器

void addChangeListener(ChangeListener l)