



第十七章 图形界面编程

讲授思路

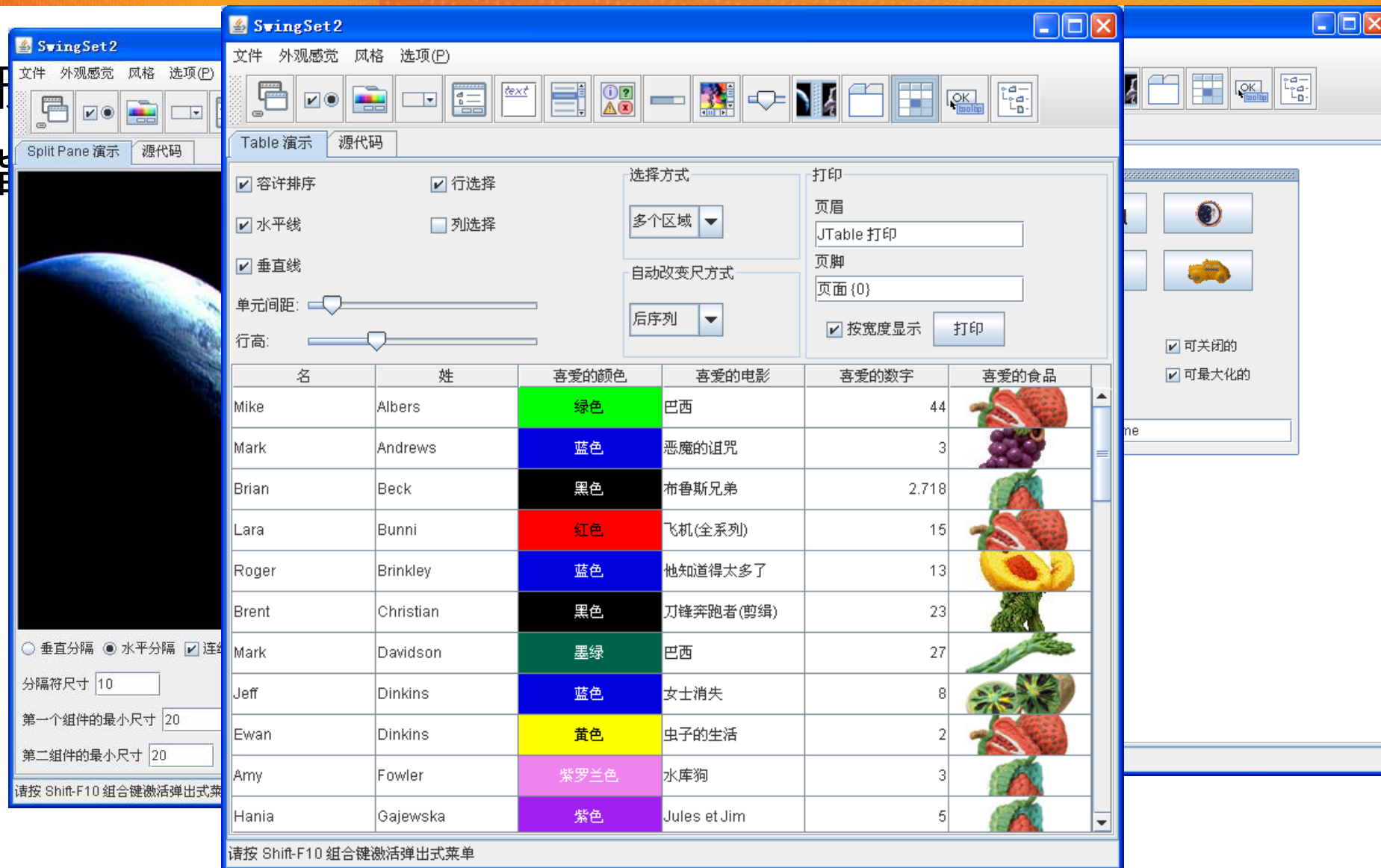
- Java GUI 编程简介
- Swing组件

讲授思路-Java GUI 编程简介

- Java图形化界面简介
- AWT
- Swing
- SWT

Java GUI 编程简介

- 图形用户接口是指



接口)

图形用户界面组成



AWT介绍

- AWT (Abstract Window Toolkit) 是Java程序提供的建立图形用户界面的工具集，主要包括容器、组件、布局管理器、事件处理模型等。
- AWT是重量级组件，通过java.awt包下的类和接口来创建GUI。

Swing简介

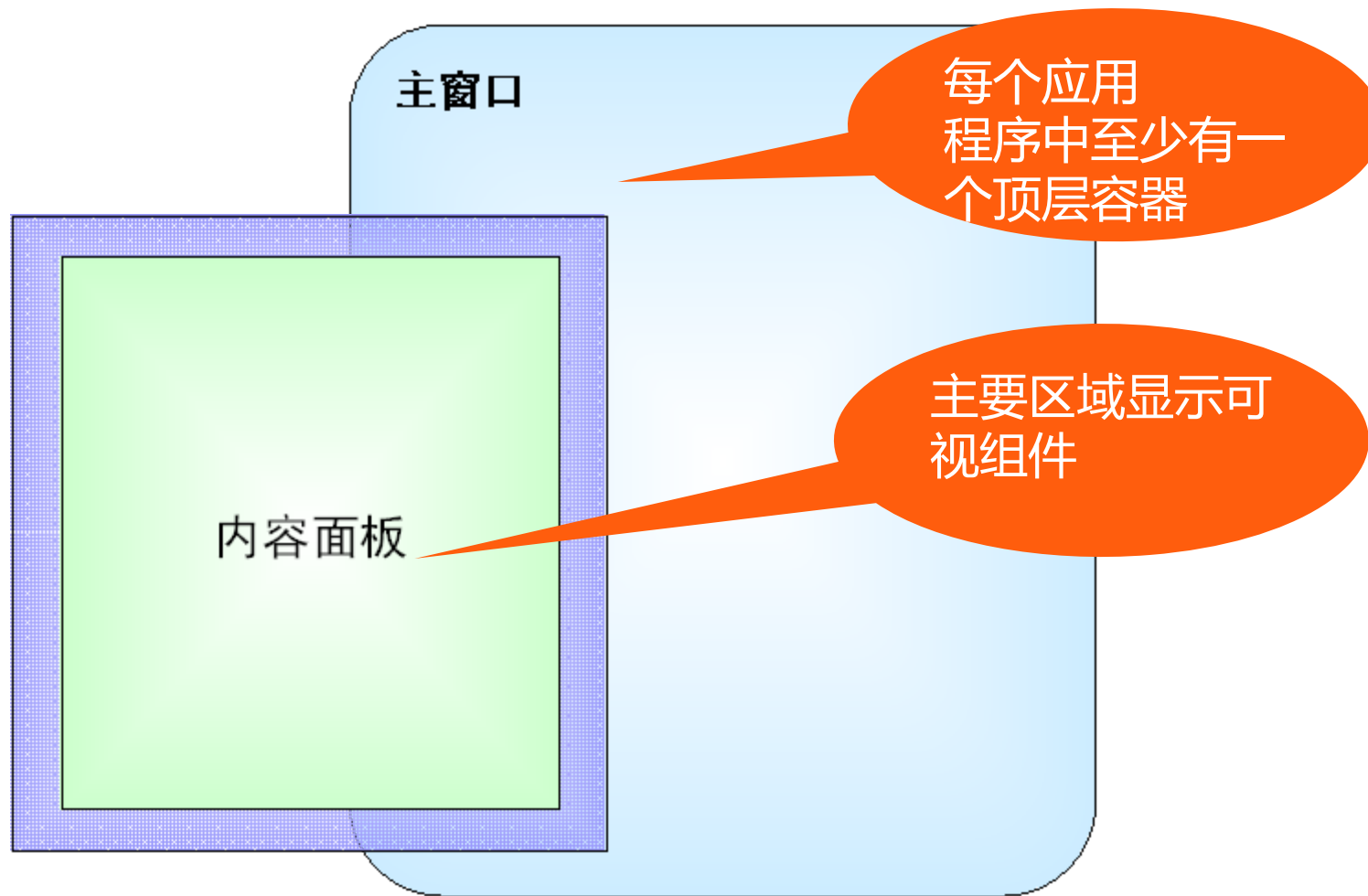
- Swing是JDK1.2版以后的轻量级组件、纯Java组件，通过javax.swing包下的类和接口来创建GUI。
- 建立在AWT基础之上，不能完全舍弃AWT。
- 可以改变外观，可以改变形状，可以利用键盘操控组件。
- Swing使用了AWT的事件模型和支持类

- SWT即是Standard Widget Toolkit。
- 最初由IBM开发的一套用于Java的图形用户界面(GUI)系统。
- 著名的开源集成开发环境 Eclipse就是用Java和SWT开发的。
- AWT+Swing 相当于 SWT。

讲授思路-Swing组件

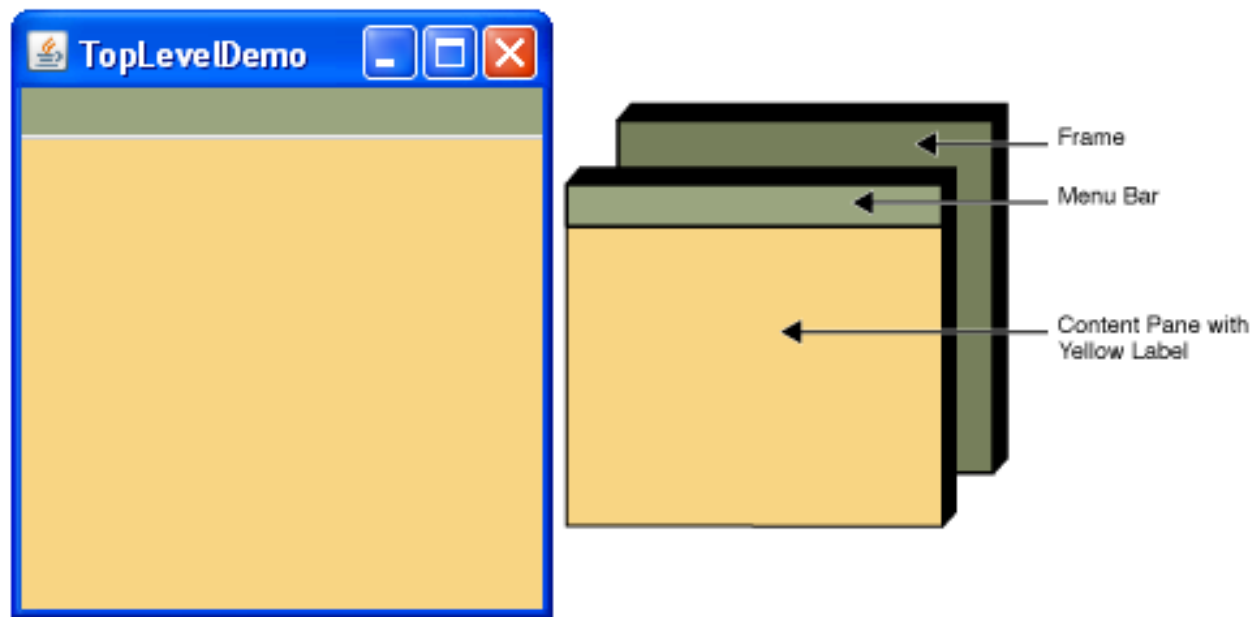
- Swing容器组件
 - JFrame
 - JPanel
- 布局管理器
- 基本常用组件的使用方法
- Swing事件处理
- Swing菜单和对话框

容器



JFrame

- 扩展自 javax.swing.JFrame 类
- 用于在 Swing 程序中创建窗口
- 包含边框、标题和用于关闭和图标化窗口的按钮



JFrame

- JFrame是Swing界面的最顶层元素(顶层容器)
- 每个JFrame都有一个与之关联的内容面板(contentPane)
- 构造方法：
 - JFrame();
 - JFrame(String title);
- 常用方法：
 - getContentPane() 返回此窗体的 contentPane 对象
 - setJMenuBar(JMenuBar menubar) 设置此窗体的菜单栏

- JFrame的setDefaultCloseOperation(int operation)方法用来决定如何响应用户关闭窗体的操作,参数有以下可选值：
 - JFrame.DO_NOTHING_ON_CLOSE : 什么也不做
 - JFrame.HIDE_ON_CLOSE : 隐藏窗体，这是JFrame的默认选项
 - JFrame.EXIT_ON_CLOSE : 结束程序

```
frame.addWindowListener(new MyWindowListener());  
class MyWindowListener extends WindowAdapter{  
    public void windowClosing(WindowEvent evt){  
        System.exit(0);  
    }  
}
```

JPanel

- JPanel为面板容器，是一个轻量级容器。
- 构造方法：
 - JPanel();
 - JPanel(LayoutManager layout);
- 常用方法：
 - add(Component component) ; 添加组件
 - getGraphics() ; 获取图形上下文
 - setBackground(Color bg); 设置背景色
 -

讲授思路-Swing组件

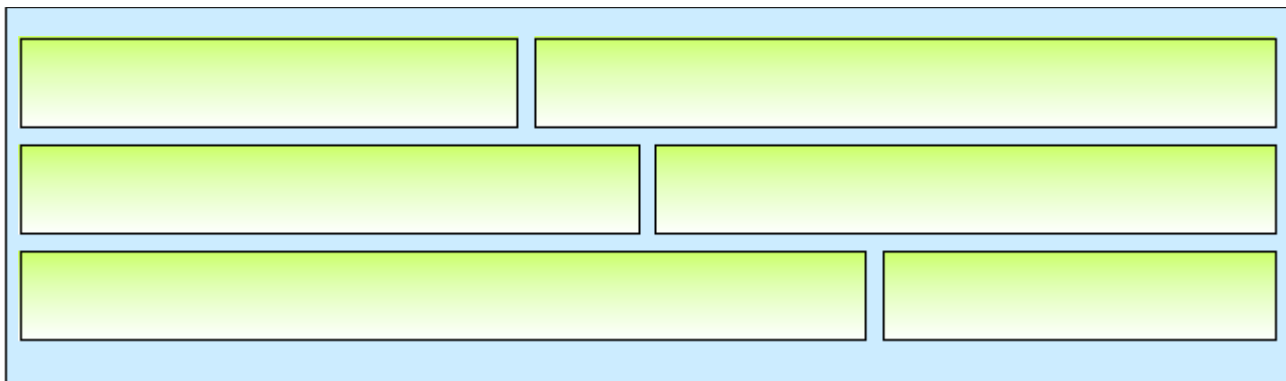
- Swing容器组件
 - JFrame
 - JPanel
- 布局管理器
- 基本常用组件的使用方法
- Swing事件处理
- Swing菜单和对话框

布局管理器

- 用户界面上的屏幕组件可以按照一定的格式布局
 - 例如水平排列，或按网格方式排列
- 常见有以下布局方式：
 - FlowLayout
 - BorderLayout
 - GridLayout

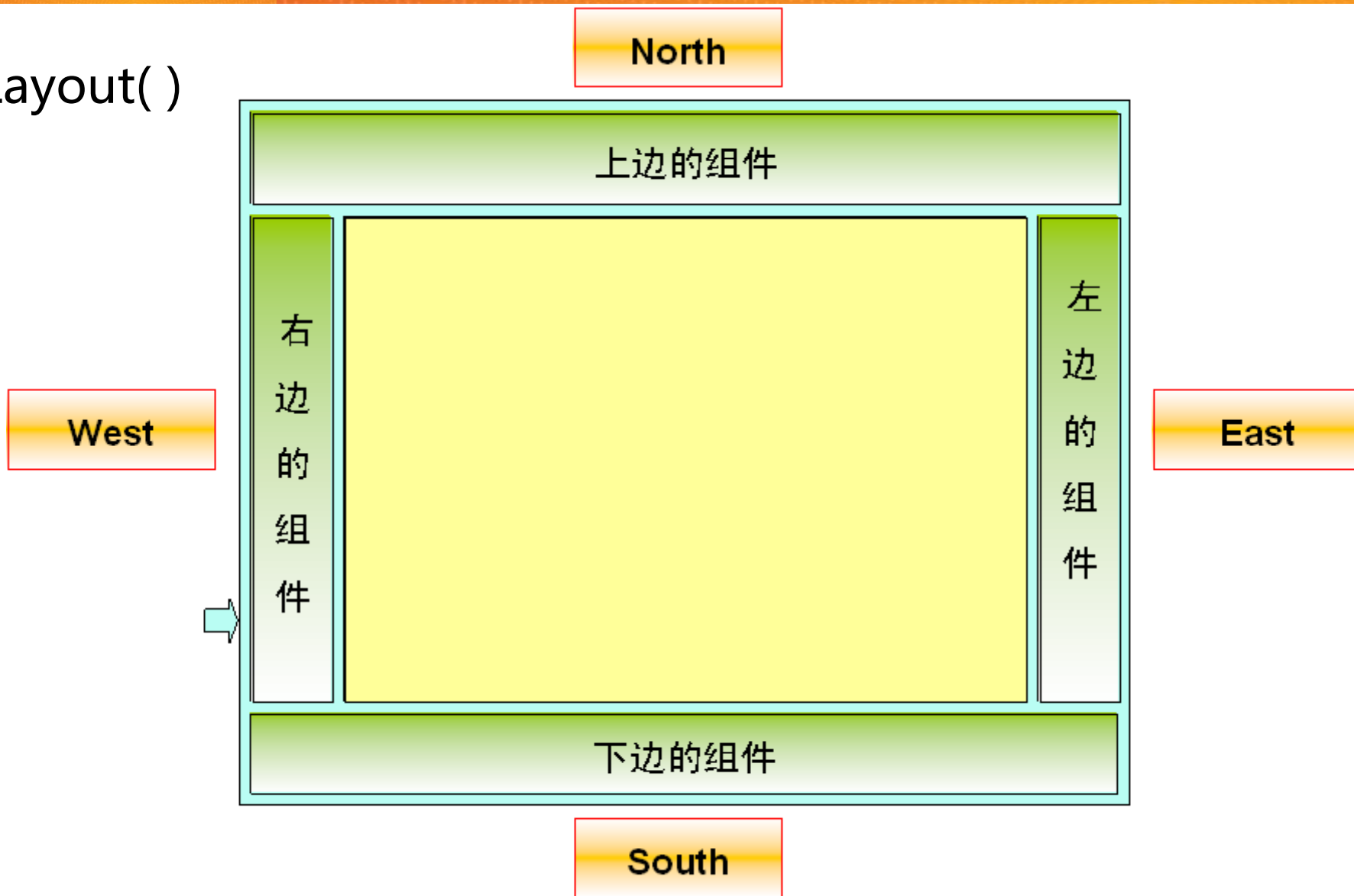
FlowLayout

- 流布局是面板的默认布局管理器
- 依次排序的组件
- FlowLayout()
 - new FlowLayout(FlowLayout.RIGHT)
 - 右对齐，默认间距5像素
 - new FlowLayout(FlowLayout.RIGHT,20,40)
 - 右对齐，水平间距20像素，垂直间距40像素



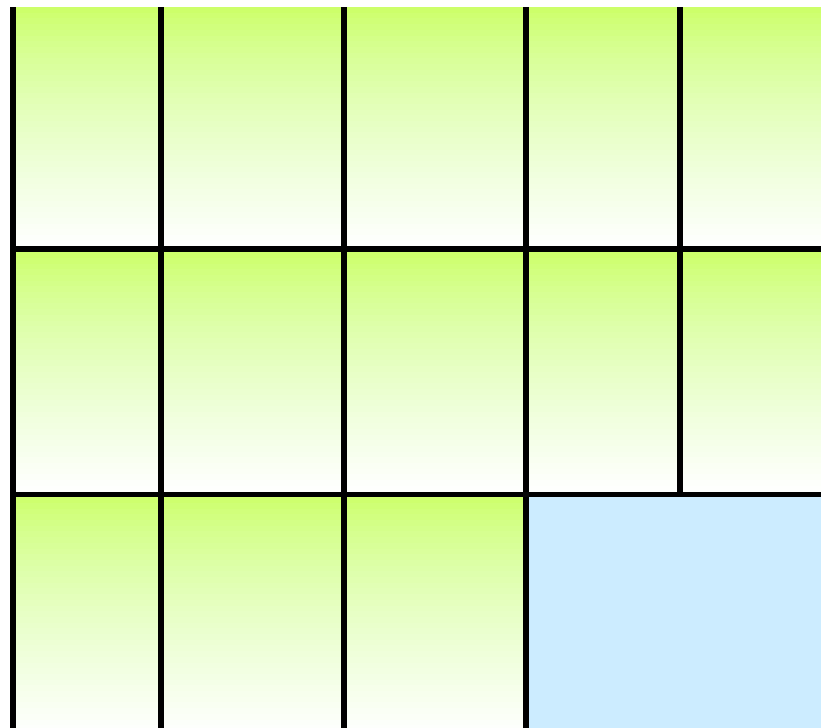
BorderLayout

- BorderLayout()



GridLayout

- 指定网格中的行数和列数，创建网格布局
- 组件大小相同
- GridLayout()
- GridLayout(int rows, int cols)



讲授思路-Swing组件

- Swing容器组件
 - JFrame
 - JPanel
- 布局管理器
- 基本常用组件的使用方法
 - 标签 (JLabel)
 - 按钮 (JButton)
 - 文本框 (JTextField、JPasswordField)
 - 文本域 (JTextArea)
 - 复选框 (JCheckBox)
 - 单选按钮 (JRadioButton)
 - 组合框 (JComboBox)
- Swing事件处理
- Swing菜单和对话框

JAVA组件库

- Java提供了AWT与Swing两套GUI组件库
 - Swing : javax.swing.* ;





- 容纳文本的组件，没有任何修饰（如：没有边缘），不能响应用户输入
- 构造方法
 - JLabel()
 - JLabel(Icon image)
 - JLabel(String text)
 - JLabel(String text, int horizontalAlignment)
 - JLabel(String text, Icon icon, int horizontalAlignment)
- 方法
 - getText()
 - setText(String text)
 - getIcon()
 - setIcon(Icon icon)

按钮—JButton



- 与用户交互使用最多的控件之一
- 构造方法
 - JButton()
 - JButton(String text)
 - JButton(Icon icon)
 - JButton(String text, Icon icon)
 - JButton(Action a) : 创建一个属性从指定的事件中获取的按钮，参数a为指定的文本
- 方法
 - getText()
 - setText(String text)
 - addActionListener(ActionListener l)
 - removeActionListener(ActionListener l)

例子

- 统计按钮被单击的次数





- 普通文本框 (JTextField)
 - 构造方法
 - JTextField()
 - JTextField(String text)
 - JTextField(String text, int columns)
 - 方法
 - addActionListener(ActionListener l)
 - removeActionListener(ActionListener l)
- 密码框 (JPasswordField)
 - 继承自JTextField
 - 具有JTextField的所有功能
 - 特有方法
 - setEchoChar(char echo) : 设置回显字符
 - getPassword()

文本域—JTextArea



- 用于输入多行文本
- 构造方法
 - JTextArea()
 - JTextArea(String text)
 - JTextArea(int rows, int columns)
 - JTextArea(String text, int rows, int columns)
- 方法
 - setRows(int rows)
 - getRows()
 - getColumns()
 - insert(String str, int pos) : 将str插入到pos ($pos \geq 0$) 位置
 - setEditable(boolean b) : 设置文本的可编辑状态
 - setLineWrap(boolean b) : 是否开启换行特性

例子

- 点击按钮，把用户名文本框和密码框的字符添加到文本域中

文本框

密码框

您输入的信息

三好学生、优秀学生干部

确定

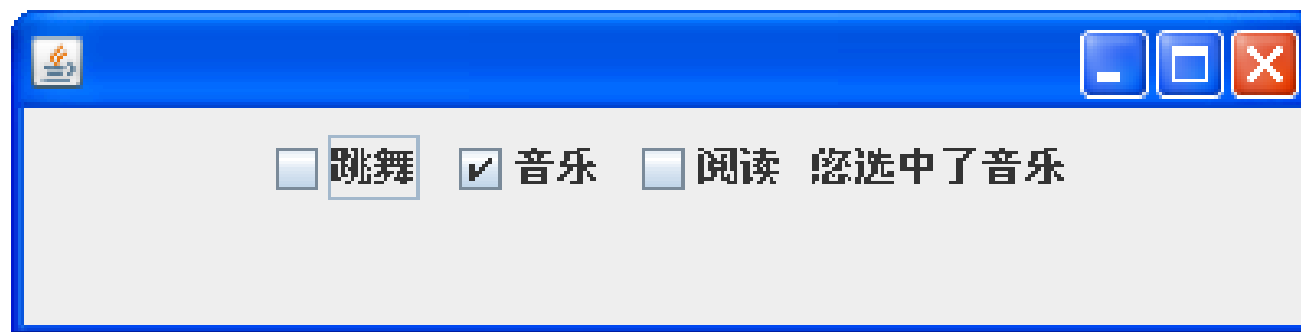
复选框—JCheckBox



- 可提供多项选择，可记录状态（是否被选中）
- 构造方法（8个）
 - JCheckBox()：创建一个无文本与图标，且未被选中的复选框
 - JCheckBox(String text)：创建指定文本，未被选中的复选框
 - JCheckBox(String text, Icon icon)：创建指定文本、图标的复选框
- 其他方法
 - isSelected()：判断该复选框是否被选中
 - setSelected(boolean state)：设置该复选框的状态(选中/不选中)
- 触发事件类型
 - ActionEvent（需要实现ActionListener接口）
 - 对应方法：addActionListener、removeActionListener
 - ItemEvent（需实现ItemListener接口）
 - 对应方法：addItemListener、removeItemListener

例子

- 爱好选择



☐ 跳舞 ☒ 音乐 ☐ 阅读 您选中了音乐

单选按钮—JRadioButton



- 提供多项选中，但只能选择一项，可记录状态(是否被选中)
- 构造方法（8个）
 - JRadioButton()：创建一个无文本与图标，且未被选中的单选按钮
 - JRadioButton(String text)：创建指定文本，未被选中的单选按钮
 - JRadioButton(String text, Icon icon)：创建指定文本、图标的单选按钮
- 其他方法
 - isSelected()：判断单选按钮是否被选中
 - setSelected(boolean state)：设置单选按钮的状态(选中/不选中)
- 触发事件类型
 - ItemEvent（需实现ItemListener接口）
 - 对应方法：addItemListener、removeItemListener
 - ActionEvent（需实现ActionListener接口）
 - 对应方法：addActionListener、removeActionListener

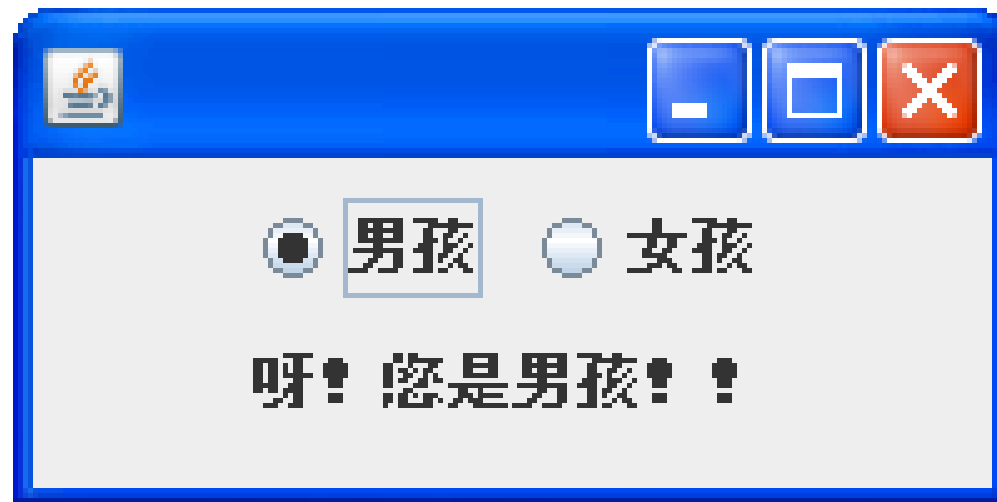
单选按钮—JRadioButton



- 注意事项
 - 需要对单选按钮进行编组 (ButtonGroup)
 - ButtonGroup的构造方法 (只有1个)
 - ButtonGroup()
 - ButtonGroup的其他方法
 - add(AbstractButton b)
 -

例子

- 性别选择



组合框—JComboBox



- 可供选择项比较多
- 类：JComboBox<E>
 - 构造方法
 - JComboBox()：没有提供选择项的组合框
 - JTextField(E[] items)：提供了选择项的组合框
 - 其他方法
 - isEditable()：判断组合框是否可供编辑
 - setEditable(boolean b)：设置组合框是否可被编辑
 - setSelectedIndex(int index)：设置默认显示指定的选项
 - setSelectedItem(E item)：设置默认显示指定的选项
 - addItem(E item)：向组合框中添加选项
 - insertItemAt(E item, int index)：将选项添加到指定位置
 - removeItemAt(int index)：删除指定位置的选项
 - removeItem(E item)：删除指定的选项
 - removeAllItems()：删除所有选项
 - removeAll()：删除所有选项，包括组合框组件本身

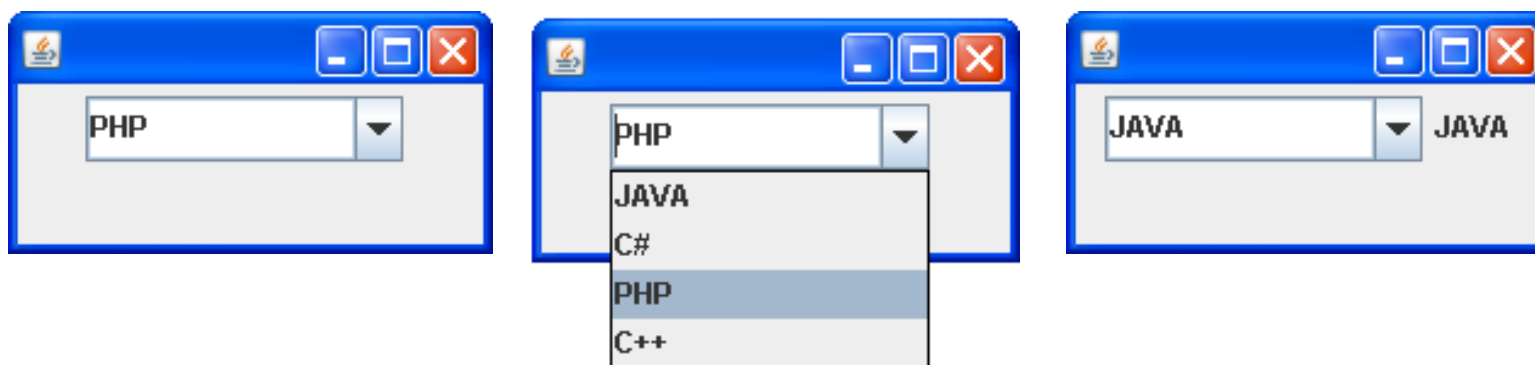
组合框—JComboBox



- 触发事件类型
 - ItemEvent (需实现ItemListener接口)
 - 使用场合：获取用户所选择的某个选项
 - 对应方法：addItemListener、removeItemListener
 - ActionEvent (需实现ActionListener接口)
 - 使用场合：用户输入或修改选项后，按下Enter键后的事件处理
 - 对应方法：addActionListener、removeActionListener

例子

- 选修课程

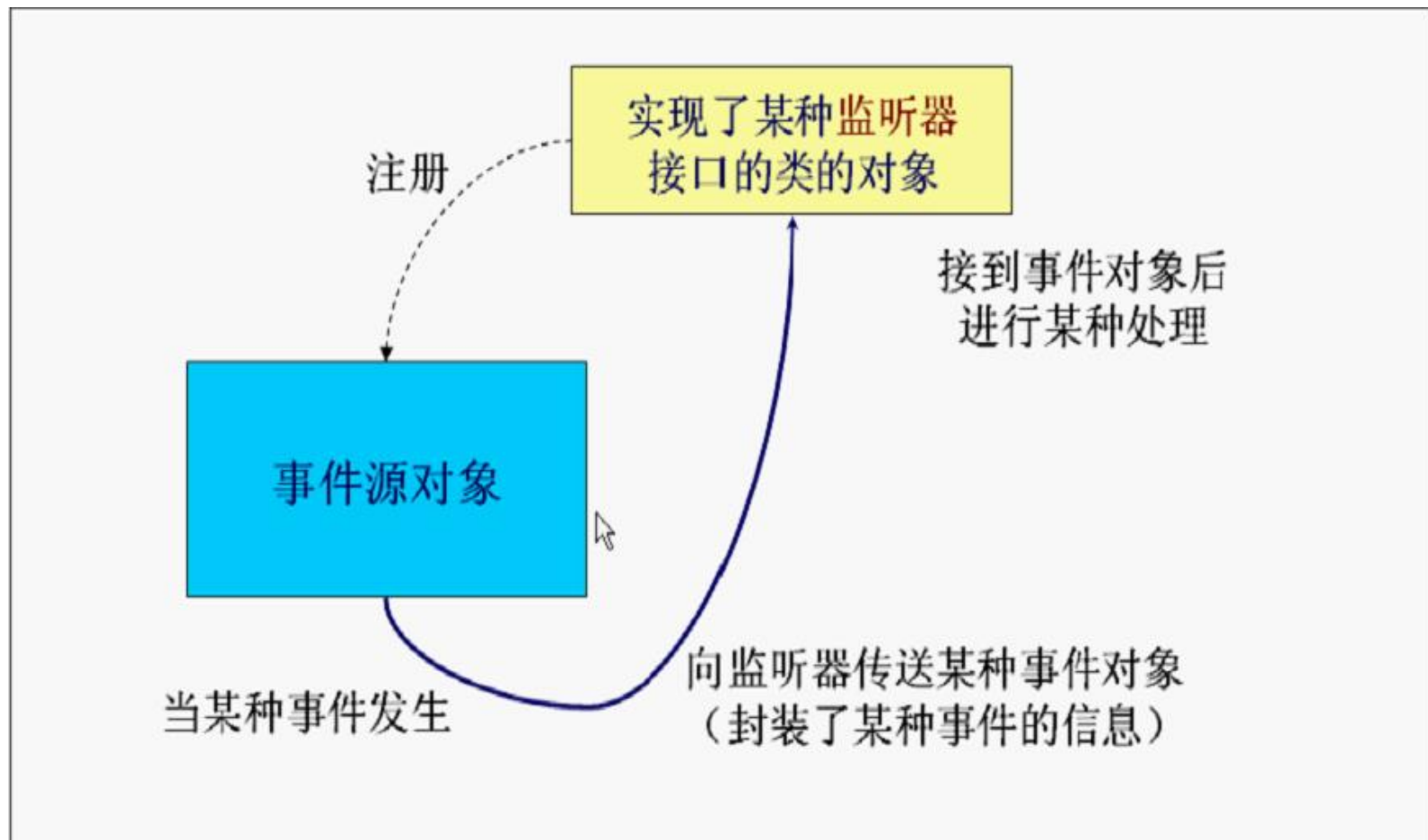


讲授思路-Swing组件

- Swing容器组件
 - JFrame
 - JPanel
- 布局管理器
- 基本常用组件的使用方法
- Swing事件处理
- Swing菜单和对话框

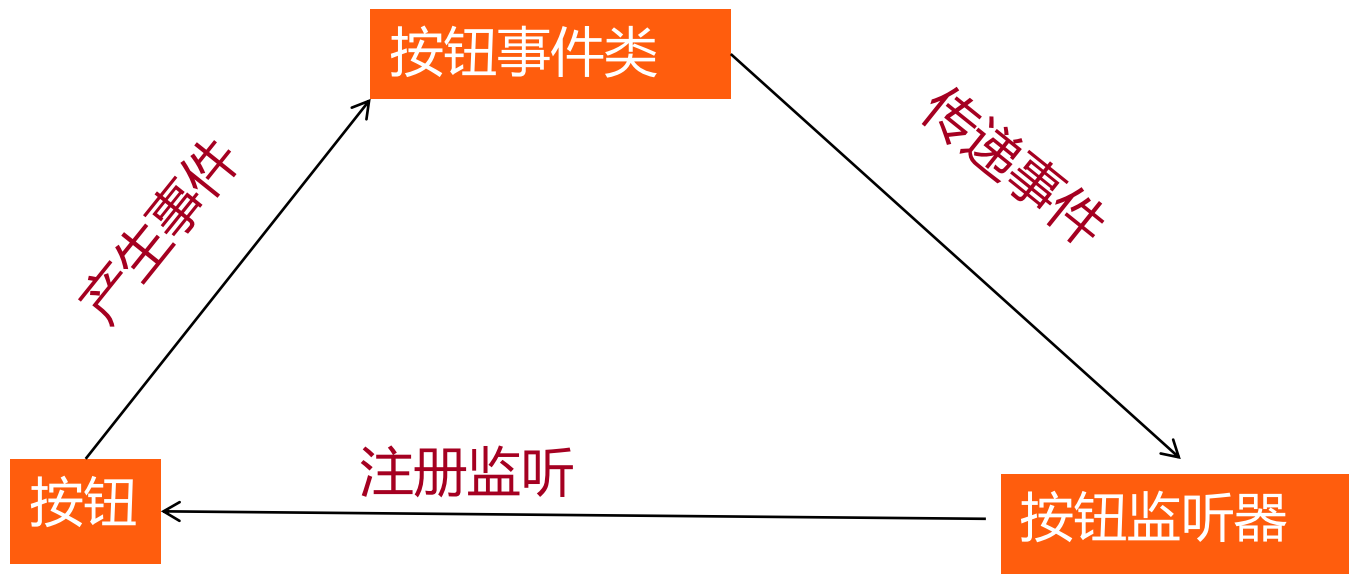
Swing事件模型原理

- 在Java开发中，对于事件的处理非常重要，比如按钮的点击、鼠标的点击、窗口的移动等等都要涉及到Java事件的应用。



事件监听

- 事件的处理步骤为：
 - 事件源上触发一个事件;比如,用户按下鼠标、按下按钮等。
 - 系统会自动产生对应的事件对象EventObject,并通知所有授权的事件监听者
 - 事件监听者中有对应的事件处理方法来处理该事件



按钮事件步骤

- 构造程序界面，创建按钮，即事件源
- 构造监听，实现处理的功能
- 注册监听
- 当用户点击按钮，即触发事件时，监听机制起作用。

讲授思路-Swing组件

- Swing容器组件
 - JFrame
 - JPanel
- 布局管理器
- 基本常用组件的使用方法
- Swing事件处理
- Swing菜单和对话框

菜单简介

- 菜单示例

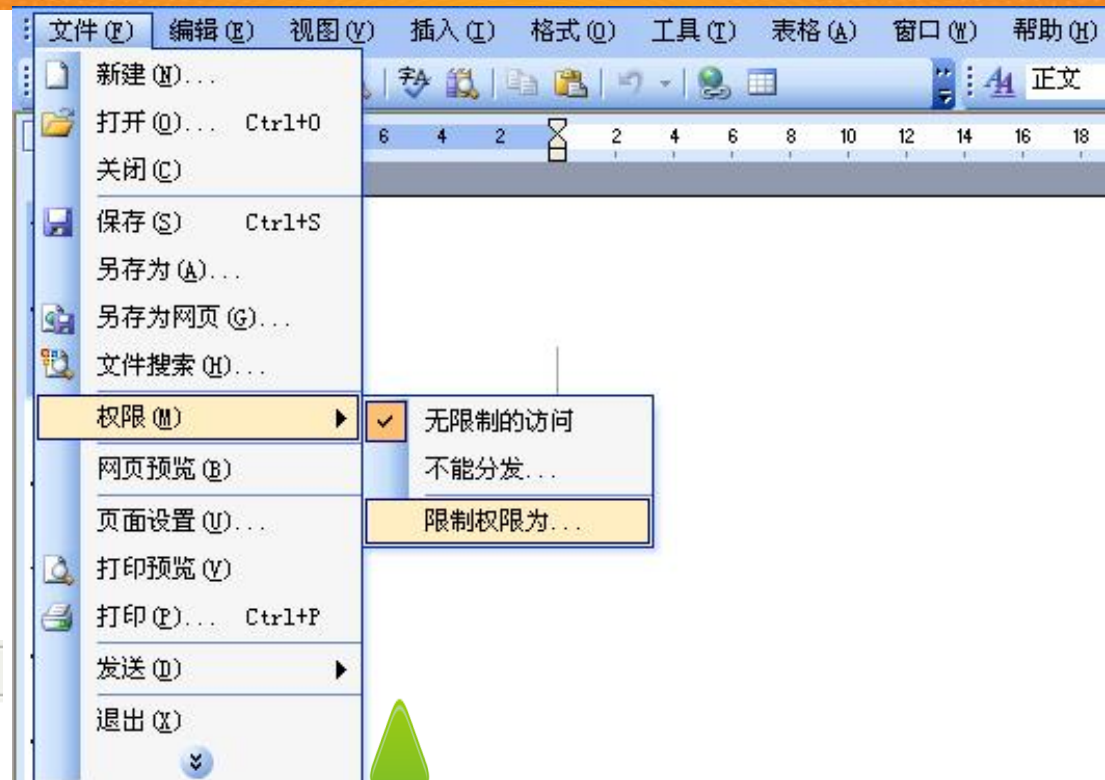
- 3级菜单

- 快捷键

- 图标

- 复选框

-

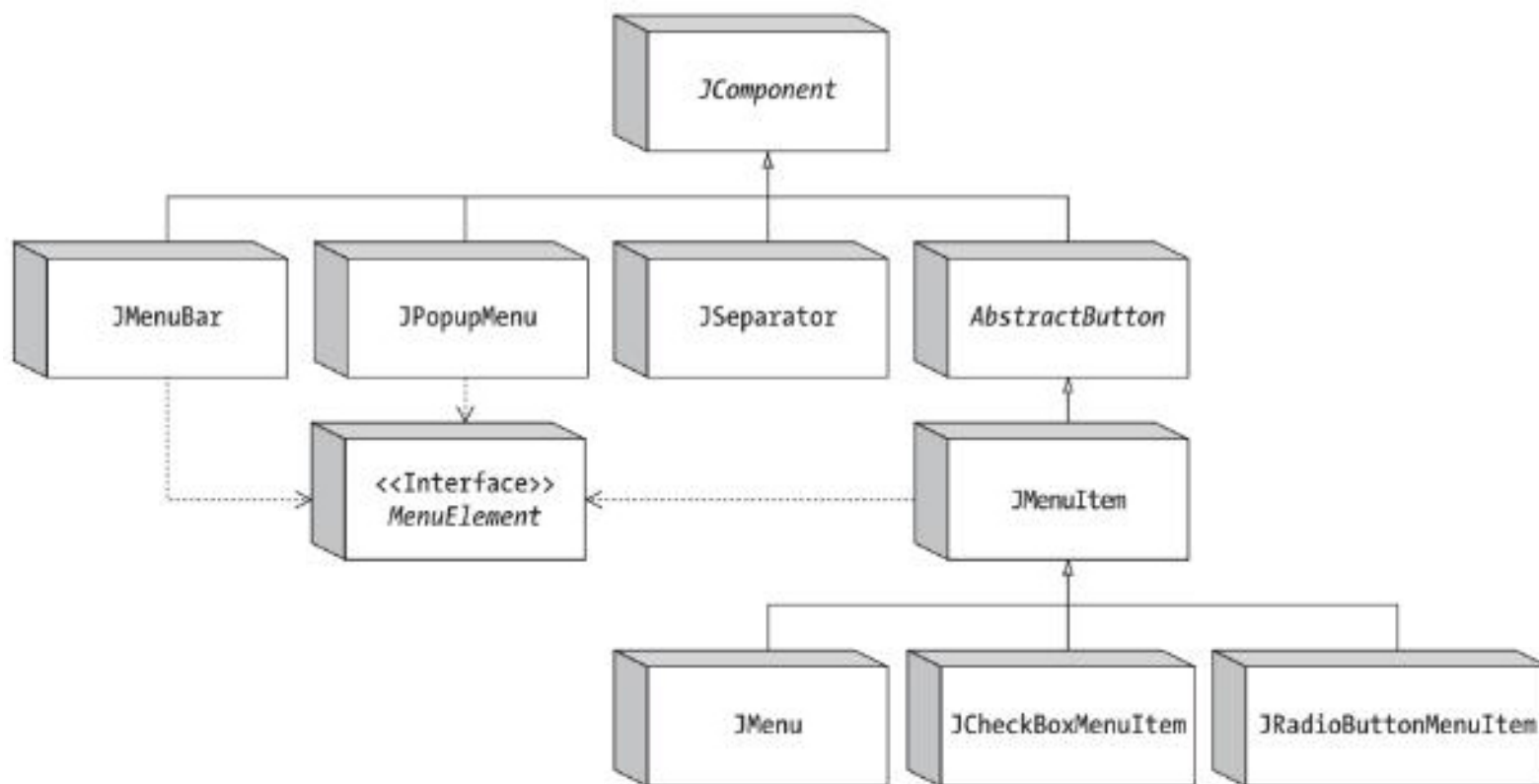


Office2003菜单栏

IE浏览器菜单栏

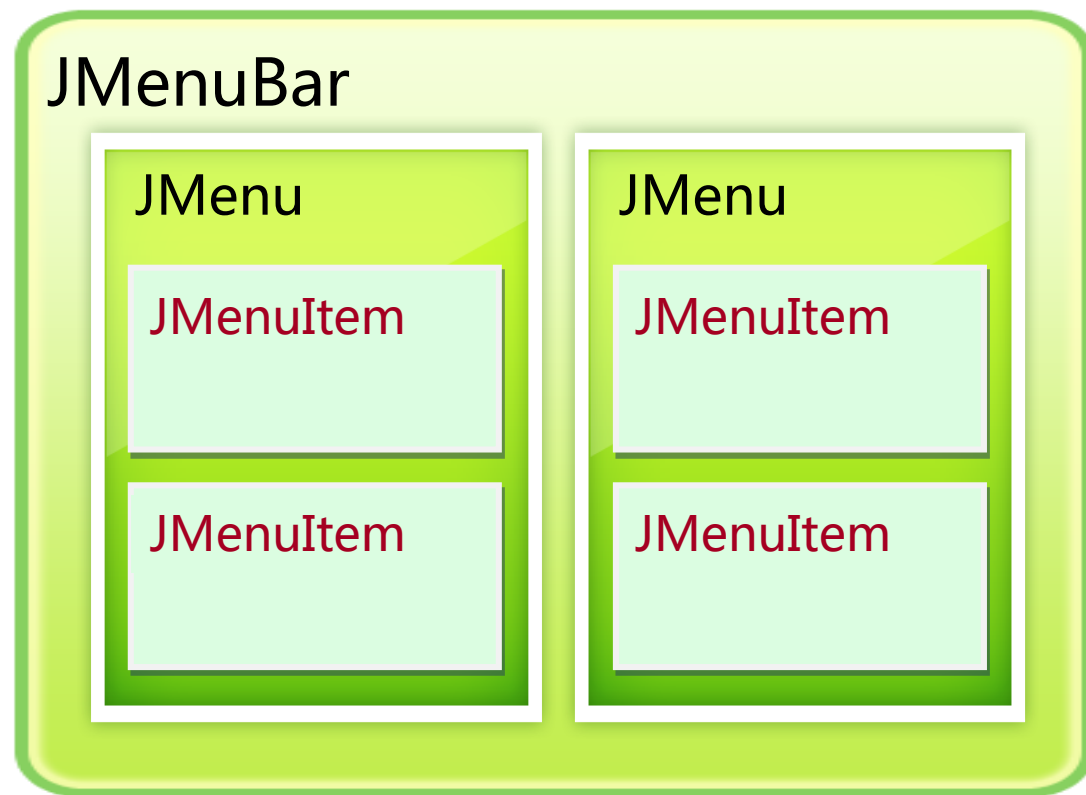
菜单简介

- 菜单组件继承自JComponent类
- 3个子类：JMenuBar、JMenu、JMenuItem



菜单简介

- JMenuBar为JMenu提供组件放置接口
- JMenu允许添加若干个JMenuItem子项



菜单子类

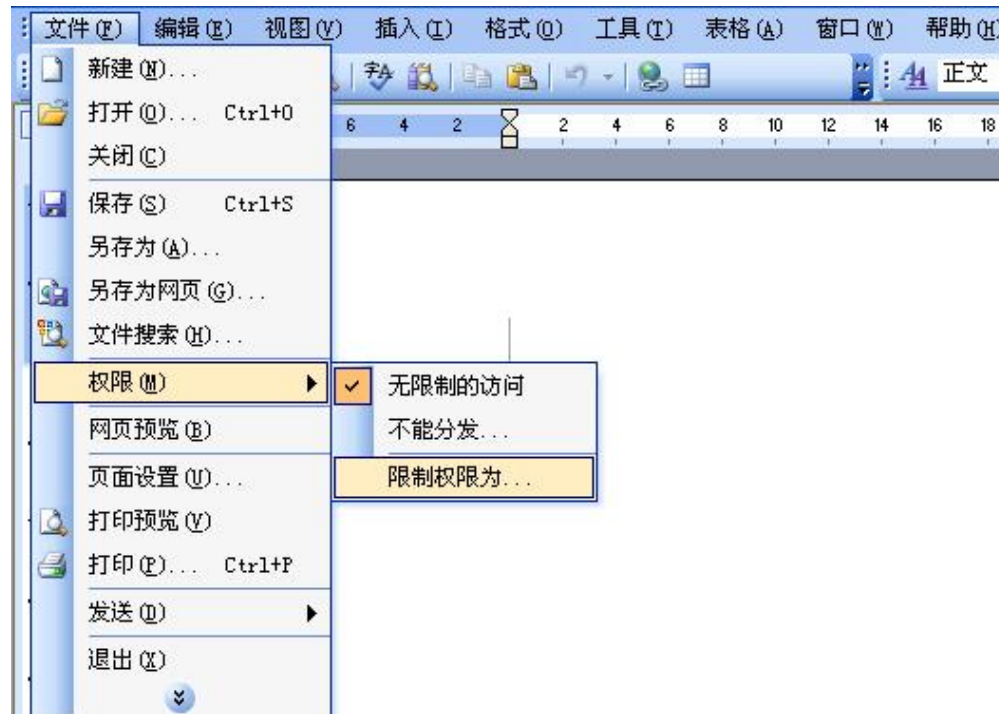
- JMenuBar
- JMenu
- JMenuItem

菜单栏—JMenuBar

- 用于实现菜单栏的组件，相当于菜单的容器
- 构造方法
 - JMenuBar()：创建一个菜单栏
- 其他方法
 - add(JMenu c)：将指定的菜单追加到菜单栏的末尾
 -
- 用途：用于放置一级菜单

菜单项—JMenuItem

- 继承自AbstractButton，相当于按钮，但不同于按钮
 - 特点
 - 鼠标经过，就认为该项菜单被选中，而不触发事件
 - 当用户在菜单项上释放鼠标，Swing也会认为该选项被选中，并触发事件完成相应的操作
- 外观包括：菜单名称、图标、快捷键、复选框



菜单项—JMenuItem

- 构造方法
 - JMenuItem() : 创建一个没有文本的菜单项
 - JMenuItem(String text) : 创建一个指定文本的菜单项
 - JMenuItem(String text, Icon icon) : 创建一个带有指定文本和图标的菜单项
 - JMenuItem(Action a) : 创建从指定 Action 获取其属性的菜单项
- 其他方法
 - setAccelerator(KeyStroke keyStroke) : 设置菜单项的快捷键
 - getAccelerator() : 获取菜单项的快捷键, 返回KeyStroke对象的引用
- 触发事件
 - 菜单事件
 - ActionEvent事件
- 用途 : 没有子菜单的菜单项

复选框菜单项（了解）

- JCheckBoxMenuItem继承自JMenuItem
- 构造方法
 - JCheckBoxMenuItem() : 创建一个没有文本或图标、且未选中的复选框菜单项
 - JCheckBoxMenuItem(String text) : 创建一个带文本、且未被选中的复选框菜单项
 - JCheckBoxMenuItem(String text, boolean b) : 创建带有指定文本和选择状态的复选框菜单项
- 其他方法
 - getState() : 返回菜单项的选定状态
 - setState(boolean b) : 设置菜单项的选定状态

单选按钮菜单项（了解）

- JRadioButtonMenuItem继承自JMenuItem
- 构造方法
 - JRadioButtonMenuItem()：创建一个没有文本或图标的单选按钮菜单项
 - JRadioButtonMenuItem(String text)：创建一个带文本的单选按钮菜单项
 - JCheckBoxMenuItem(String text, boolean b)：创建一个具有指定文本和选择状态的单选按钮菜单项
- 注意事项
 - 给单选按钮菜单项编组（ButtonGroup）
 - 构造方法：ButtonGroup()：创建一个组
 - 其他方法：add(AbstractButton b)：将按钮添加到组中

菜单—JMenu

- 继承自JMenuItem
- 既可用作一级菜单，又可以作为子菜单添加到其他菜单中



菜单—JMenu

- 构造方法
 - JMenu() : 创建一个没有文本的菜单
 - JMenu(String s) : 创建一个指定文本的菜单
 - JMenu(Action a) : 创建一个从指定Action获取属性的菜单
 - JMenu(String s, boolean b) : 创建一个具有指定文本的菜单，并且设置该菜单是否为分离式菜单
- 其他方法
 - add(String s) : 创建指定文本的菜单，并追加到此菜单的末尾
 - add(Component c) : 将指定组件追加到此菜单的末尾
 - addSeparator() : 在此菜单的末尾添加一个分隔线
 - remove(JMenuItem) : 从此菜单中移除指定的菜单项

菜单—JMenu

- 触发事件
 - ActionEvent
 - MouseEvent
 - 特殊事件：菜单事件
 - addMenuListener (实现menuCanceled、menuDeselected、menuSelected方法)
- 用途：含有子菜单的菜单项

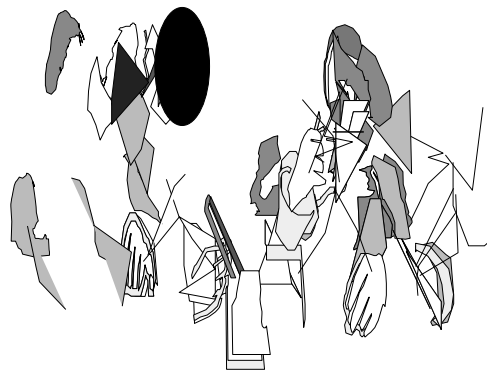
菜单的使用方法

- 下拉式菜单
- 弹出式菜单



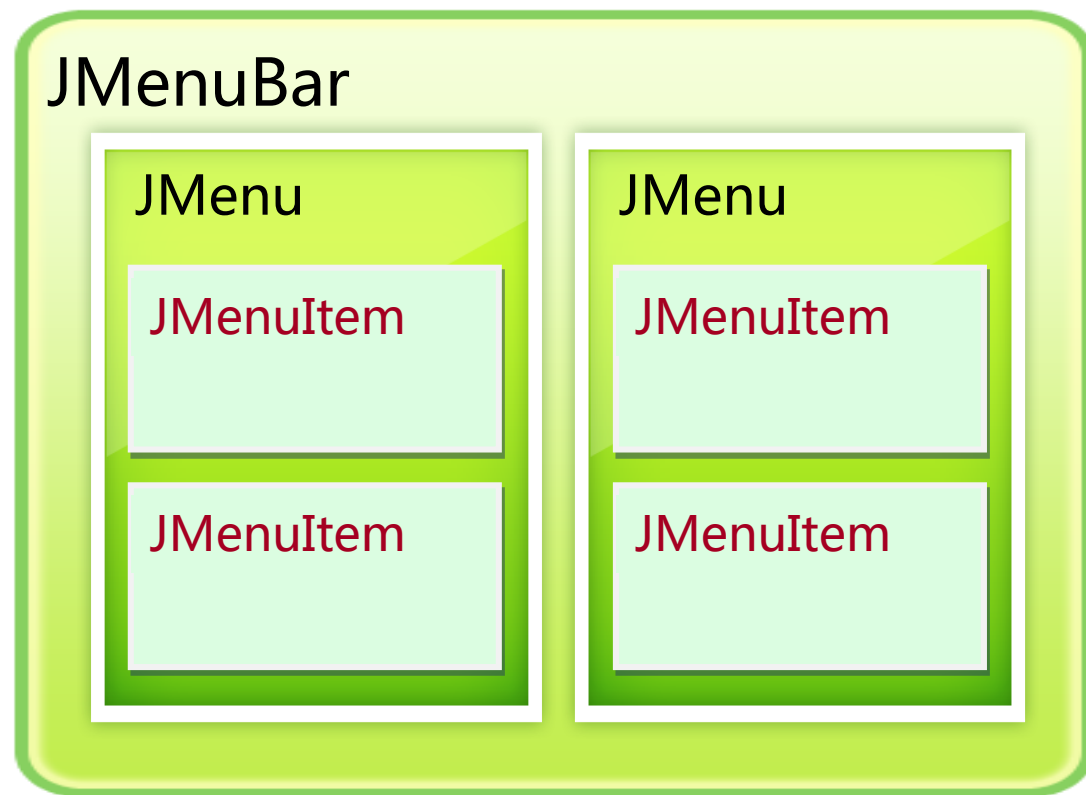
思考

- 菜单组成元素有哪些？
- 预计大概步骤有哪些？
- 实现特定的功能应该做什么操作？



菜单结构

- JMenuBar为JMenu提供组件放置接口
- JMenu允许添加若干个JMenuItem子项



下拉式菜单

- 位于窗口顶部
- 点击某一个名字会打开一个菜单列表



下拉式菜单

- 主要步骤

- 创建菜单栏
 - 没有子菜单的菜单用JMenuItem
 - 有子菜单的菜单使用JMenu
- 给菜单添加子菜单
- 给窗体添加菜单栏
- 给菜单添加事件监听器

```
JMenuBar menuBar= new JMenuBar();  
JMenu menuFile = new JMenu( "文件" );  
JMenuItem menuItemOpen = new JMenuItem( "打开" );  
JMenu menuNew = new JMenu( "新建" );  
JMenuItem menuItemTxt = new JMenuItem( "TEXT" );  
menuBar.add(menuFile);  
menuFile.add(menuNew);  
menuFile.add(menuItemOpen);  
menuNew.add(menuItemTxt);  
this. setJMenuBar(menuBar);
```

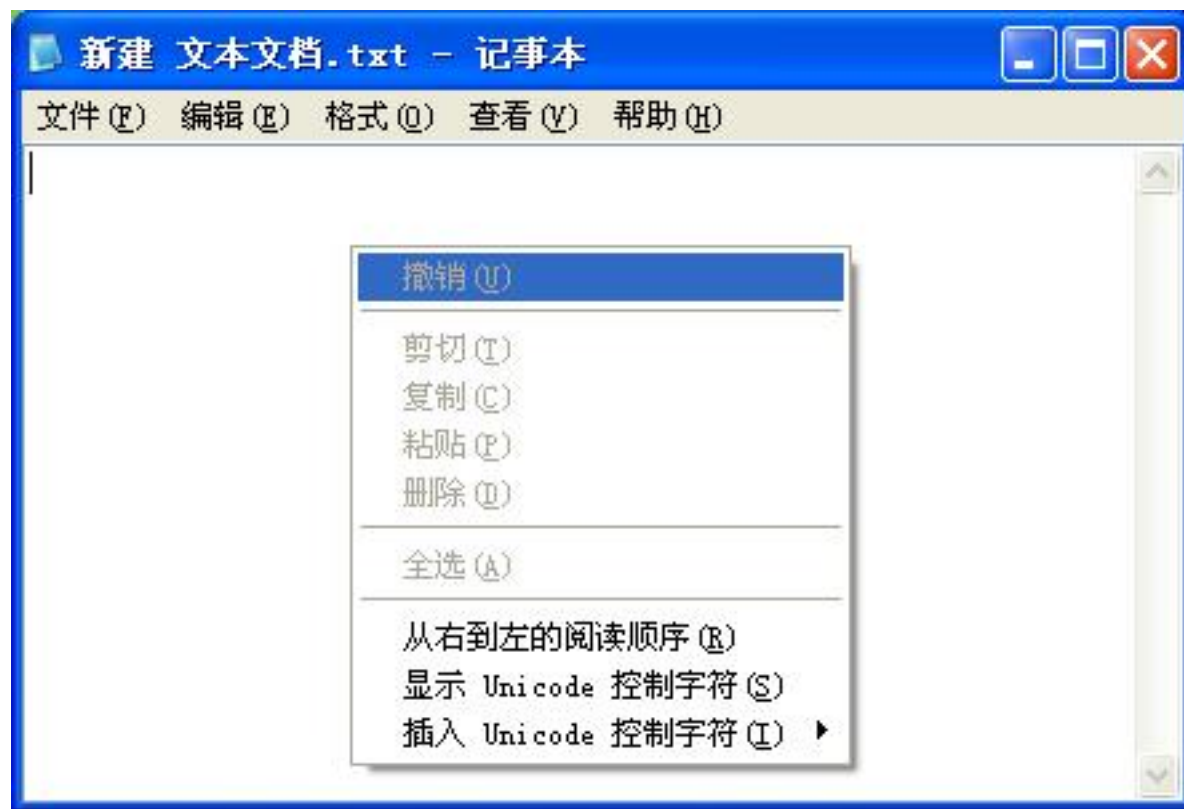

下拉式菜单

- 事件
 - 点击“退出”菜单项，关闭程序
 - 触发ActionEvent事件
 - 给“退出”菜单项注册ActionListener事件监听器

```
//给”退出“菜单项注册事件监听器  
menuItemExit.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        System.exit(0);  
    }  
});
```

弹出式菜单

- 单击鼠标右键时弹出菜单列表
- 菜单位置不固定

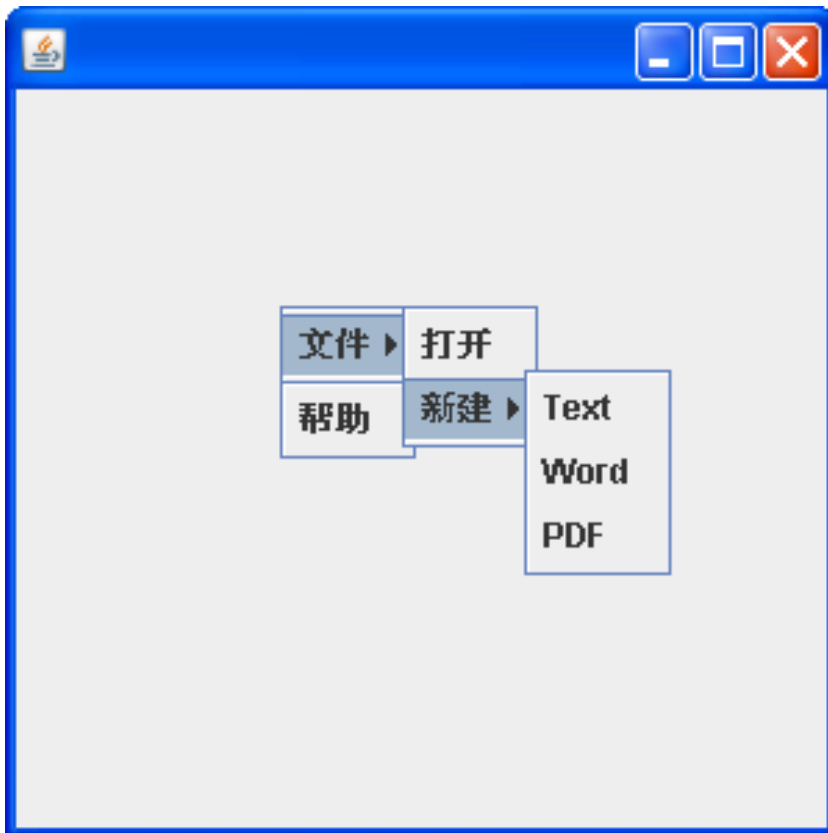


弹出式菜单

- JPopupMenu
- 构造方法（2个）
 - JPopupMenu()：构造一个不带“调用者”的弹出式菜单
 - JPopupMenu(String label)：构造一个具有指定标题的弹出式菜单
- 其他方法
 - add(JMenuItem menuItem)：将指定菜单项添加到此菜单的末尾
 - add(String s)：创建具有指定文本的菜单项，并将其添加到此菜单的末尾
 -
- 弹出菜单项触发事件
 - ActionEvent

弹出式菜单

- 特别注意：必须调用show方法，菜单才能显示出来
 - `show(Component invoker, int x, int y)`



分析菜单组成：

1. 2个一级菜单
文件：JMenu
帮助：JMenuItem
2. 2个二级菜单
打开：JMenuItem
新建：JMenu
3. 3个三级菜单

弹出式菜单

- 主要步骤

- 创建弹出式菜单

- 创建菜单及子菜单

- 没有子菜单的菜单用JMenuItem

- 有子菜单的菜单使用JMenu

- 给菜单添加子菜单

- 给组件添加鼠标事件

- 调用show方法

```
JPopupMenu popupMenu = new JPopupMenuJMenuBar();
JMenu menuFile = new JMenu( "文件" );
JMenuItem menuItemOpen = new JMenuItem( "打开" );
JMenu menuNew = new JMenu( "新建" );
JMenuItem menuItemTxt = new JMenuItem( "TEXT" );
popupMenu.add(menuFile);
menuFile.add(menuNew);
menuFile.add(menuItemOpen);
menuNew.add(menuItemTxt);
```

弹出式菜单

- 事件

- 点击鼠标右键弹出菜单项

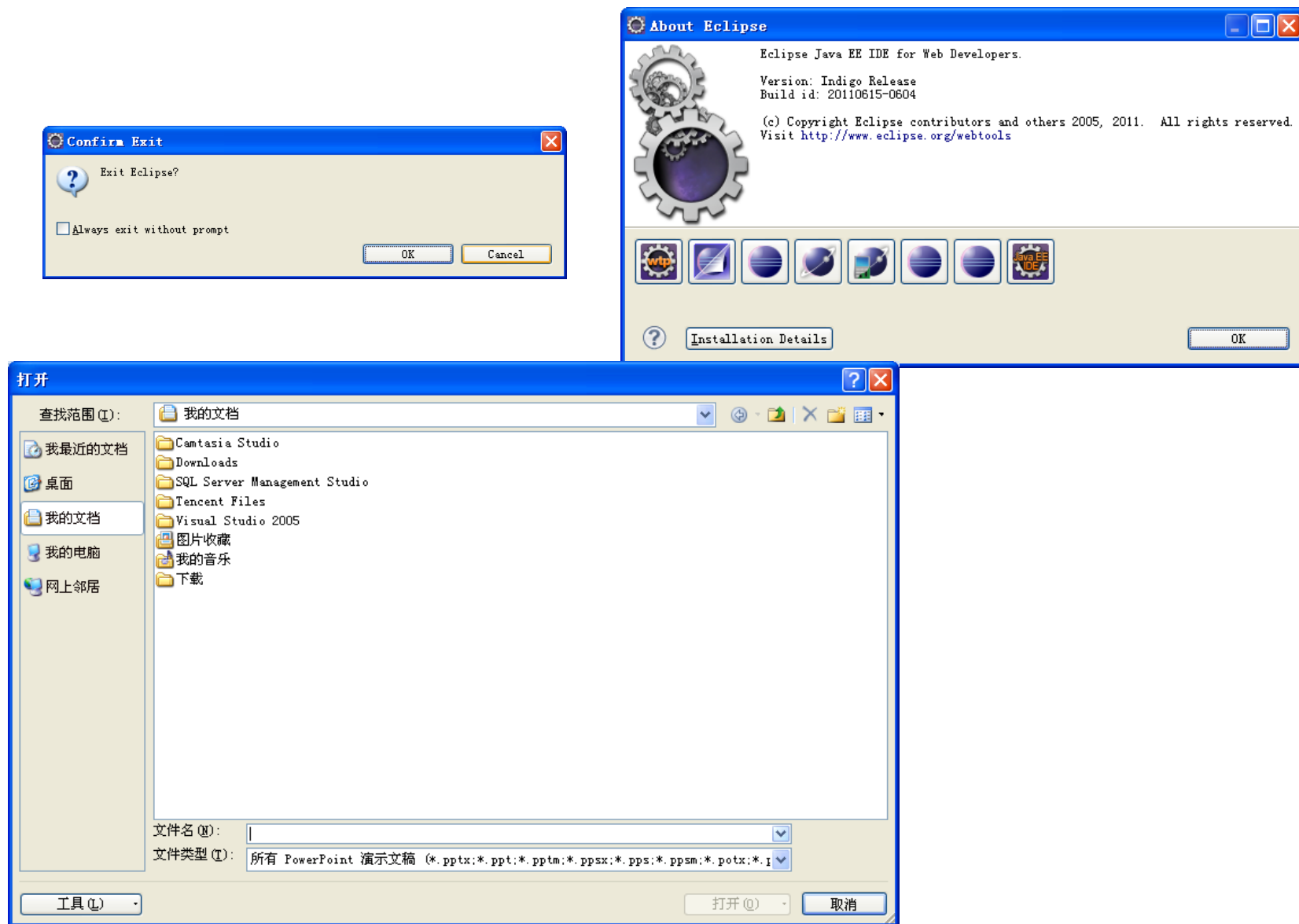
- 触发JPanel容器的鼠标事件 (addMouseListener)

- 实现mousePressed方法
 - 注意必须调用show方法显示弹出式菜单

```
//给JPanel实例对象panel容器注册事件监听器
panel.addMouseListener(new MouseListener() {
    @Override
    public void mousePressed(MouseEvent e) {
        int mods=e.getModifiers();
        if((mods&InputEvent.BUTTON3_MASK)!=0){
            //调用show方法显示弹出式菜单
            popupMenu.show(panel,e.getX(),e.getY());
        }
    }
});
```


对话框简介

- 对话框



对话框分类

- 模式对话框
 - 模式对话框创建后，程序的其他窗口便不能进行操作，必须将该窗口关闭后，其他窗口才能进行操作。
- 非模式对话框
 - 非模式对话框则无需这样，它不强制要求用户立即反应。

讲授思路

- 对话框简介
- JOptionPane

JOptionPane

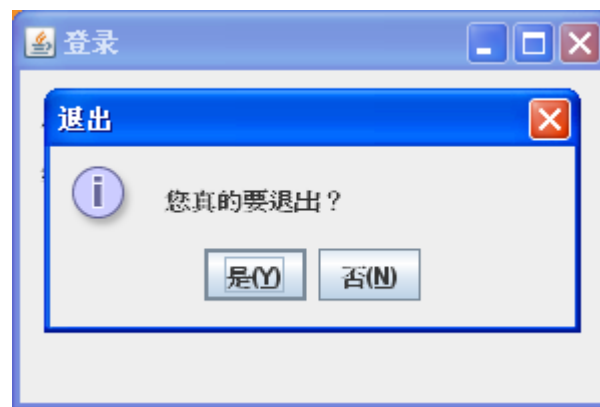
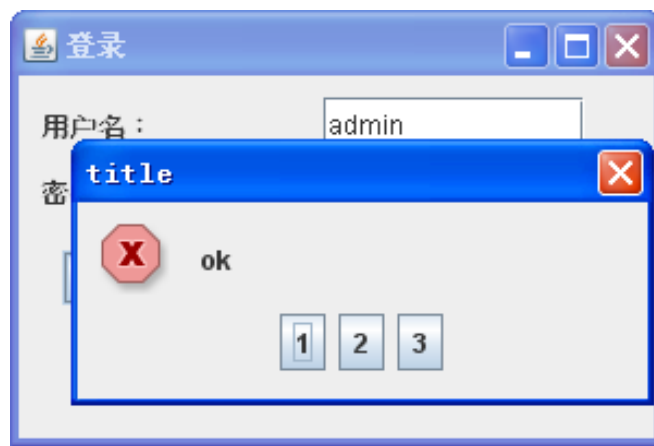
- JOptionPane类中的提供了若干个static方法来生成各种标准的对话框。
- 包括：
 - `ConfirmDialog` --- 确认对话框，提出问题，然后由用户自己 来确认（按"Yes"或"No"按钮）
 - `InputDialog` --- 提示输入文本
 - `MessageDialog` --- 显示信息
 - `OptionDialog` - - 组合其它三个对话框类型。
- 这些对话框都是模式对话框

对话框

- showXXXDialog(Component parentComponent,
Object message,
String title,
int optionType,
int messageType)
- 参数：
 - parentComponent - 确定在其中显示对话框的 Frame
 - message - 要显示的内容
 - title - 对话框的标题字符串
 - optionType - 指定可用于对话框的选项的整数：
YES_NO_OPTION 或 YES_NO_CANCEL_OPTION
 - messageType - 指定此消息种类的整数；主要用于确定来自可插入外观的图标：
ERROR_MESSAGE、INFORMATION_MESSAGE、WARNING_MESSAGE、
QUESTION_MESSAGE 或 PLAIN_MESSAGE

例子

- 登录窗体



总结

- Java GUI 编程简介
 - Java图形化界面简介
 - AWT
 - Swing
 - SWT
- Swing组件
 - Swing容器组件
 - 布局管理器
 - 基本常用组件的使用方法
 - Swing事件处理
 - Swing菜单和对话框



Thank You