Java技术

第五章（1）

Java GUI设计与事件处理

路　强

luqiang@hfut.edu.cn

合肥工业大学计算机与信息学院

本章学习提示

- 本章学习Java语言的用户界面的设计，以及如何让用户和程序进行友好方便的交互。
  - Java中AWT包
  - 布局管理器的概念和使用
  - Swing包的简介
  - Java的事件处理机制

目　录

1 AWT 组件

2 布局管理器

3 Swing 包

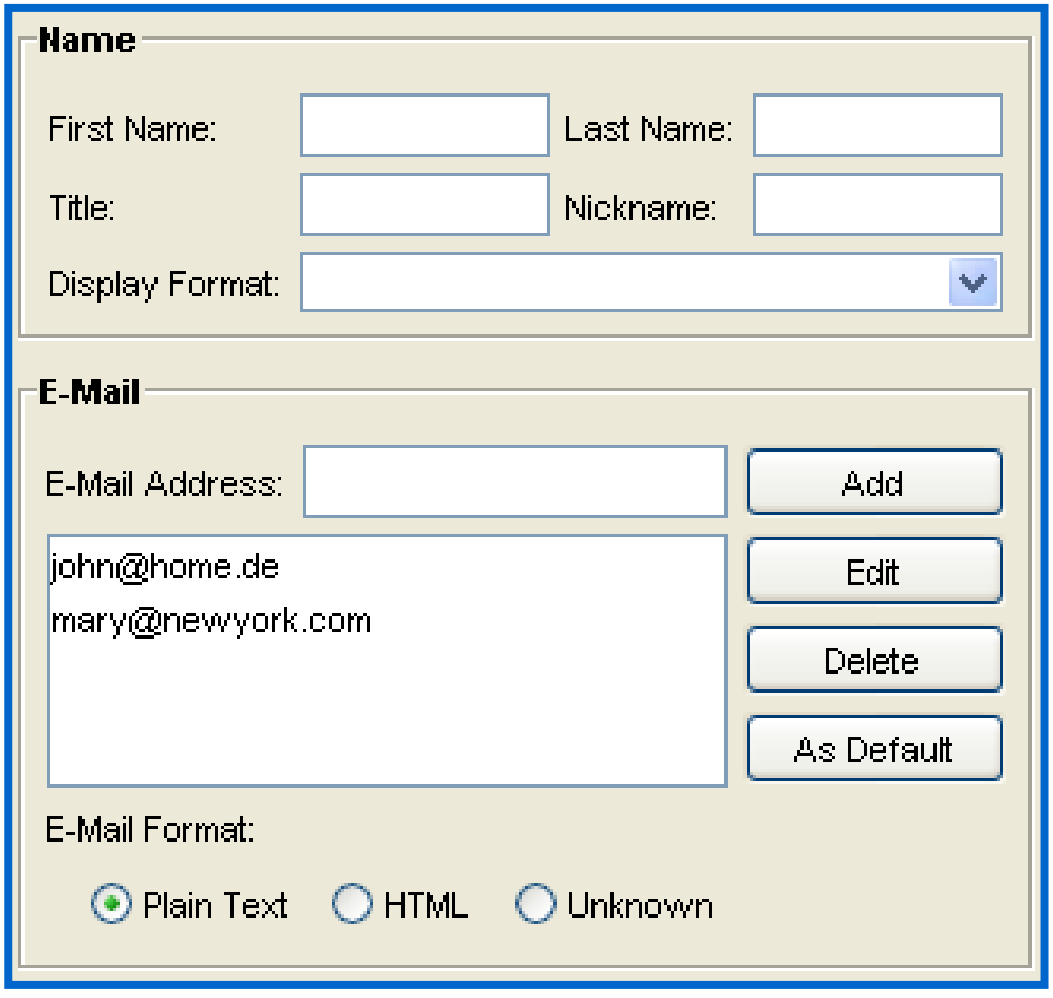
Java 的图形用户界面AWT包

- AWT：Abstract Window Toolkit
- GUI： Graphical User Interface

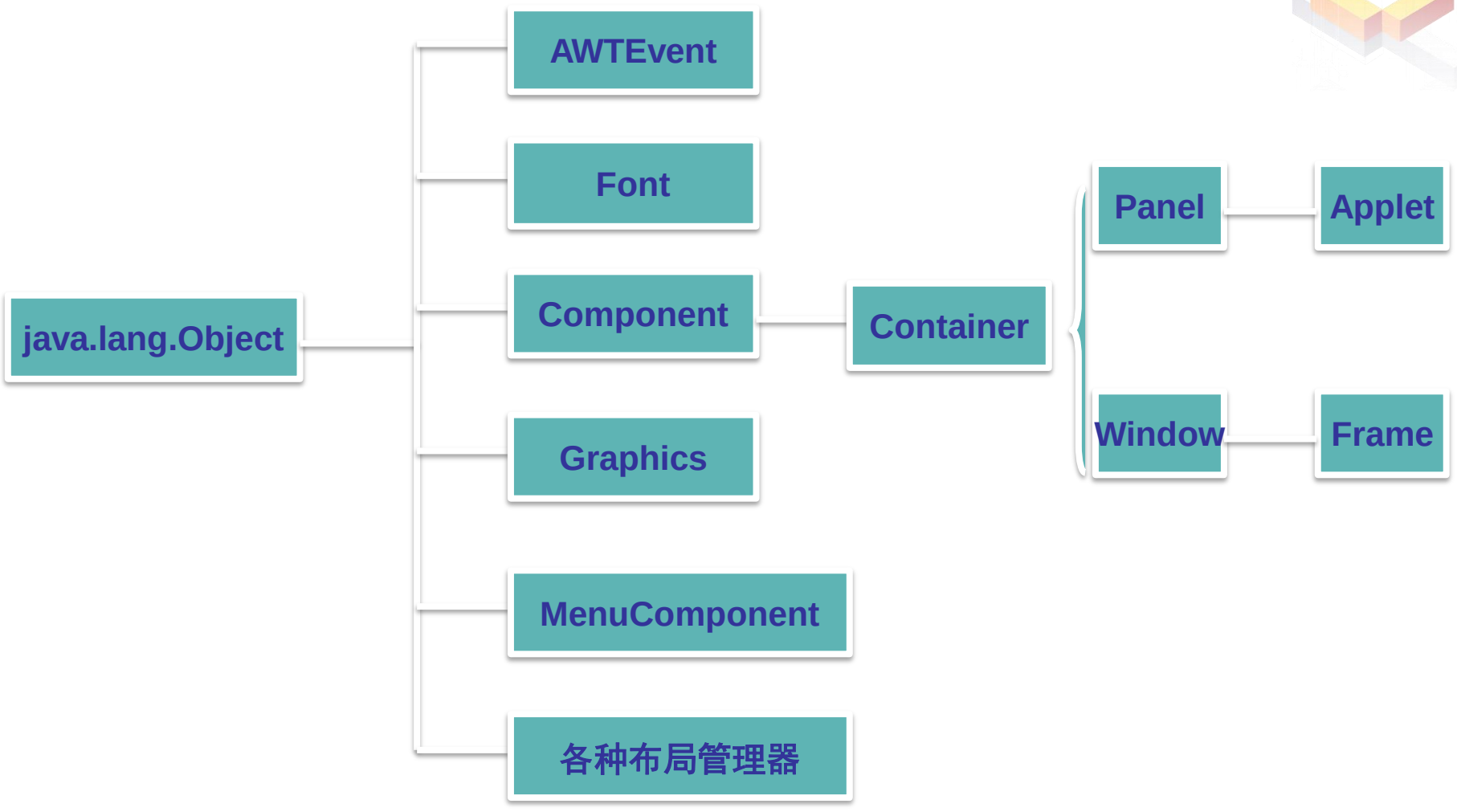
通过提供菜单、按钮、标签标识、鼠标等，完成对计算机发出指令、启动应用程序等操作任务。

- 在Java中，AWT是用来处理图形最基本的方式，它可以用来创建Java的applet和窗口。AWT包提供：
  - 图形界面组件：如：窗口、按钮、菜单等
  - 容器 ： 是GUI元素的容器。实现组件管理、布局管理
  - 布局管理器：用来安排图形界面组件的位置
  - Graphics：在组件上进行图形绘制所使用的图形环境的父类
  - 事件处理对象：用来处理图形界面组件所触发的事件

GUI之例


Name
First Name:
Last Name:
Title:
Nickname:
Display Format:
E-Mail
E-Mail Address:
Add
john@home.de
mary@newyork.com
Edit
Delete
As Default
E-Mail Format:
Plain Text
HTML
Unknown

java.awt包


java.lang.Object
AWTEvent
Font
Component
Container
Panel
Applet
Window
Frame
Graphics
MenuComponent
各种布局管理器

常用容器

- **组件**：是GUI中最基本的组成部分，是一个可以图形化显示的，可以与用户进行**交互对象**。如：按钮、文本框。
- **容器**：就是**组件放置的地方**。其本身可以看成是一个特殊的组件，只不过可以容纳其他的组件或容器。

建立窗口的Frame类

- Java 中“窗口”被视为一个容器。它可以把各种不同的图形界面组件放置到这个容器中，而这些图形界面组件就是容器中的接口，通过图形界面组件所提供的方法，来完成一定的功能。
- 创建窗口的基本和必要操作包括如下步骤：
  1. 给窗口一个标题
  2. 设置窗口的大小
  3. 在屏幕的某个位置放置窗口
  4. 显示窗口

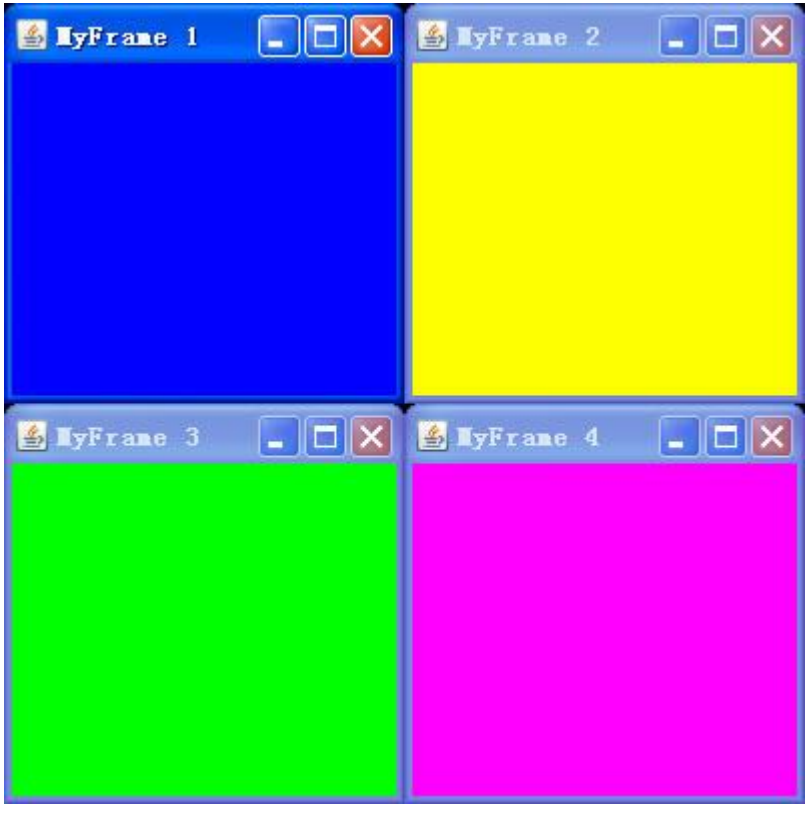
显示窗体之例

```
1.  //show GUI Frame
2.  import java.awt.*;
3.  class myFrame extends Frame {//新建的窗口类
4.          myFrame(String s){
5.              super(s);
6.          this.setSize(300,200);
7.          this.setBackground(Color.yellow);
8.          this.setVisible(true);
9.      }
10. }
11. public class HelloGuiWorld01{
12.     public static void main(String[] args){
13.         myFrame frm=new myFrame("HelloGUIWorld!");
14.     }
15. }
```

Frame类主要方法

- Component类是所有GUI对象的祖先，Window类是Frame类的父类。
- 对于Frame外观的操作比较重要的方法如下：
  - setTitle——设置窗口中标题栏的文字。
  - setResizable——设置用户是否可以改变框架大小。
  - dispose ——关闭窗口，并回收该窗口的所有资源。
  - setSize——设置组件的大小。
  - setBackground——设置组件的背景颜色。
  - setVisible——设置组件是否可见。
  - setBounds——重新设置组件的大小和位置。

Frame窗口例



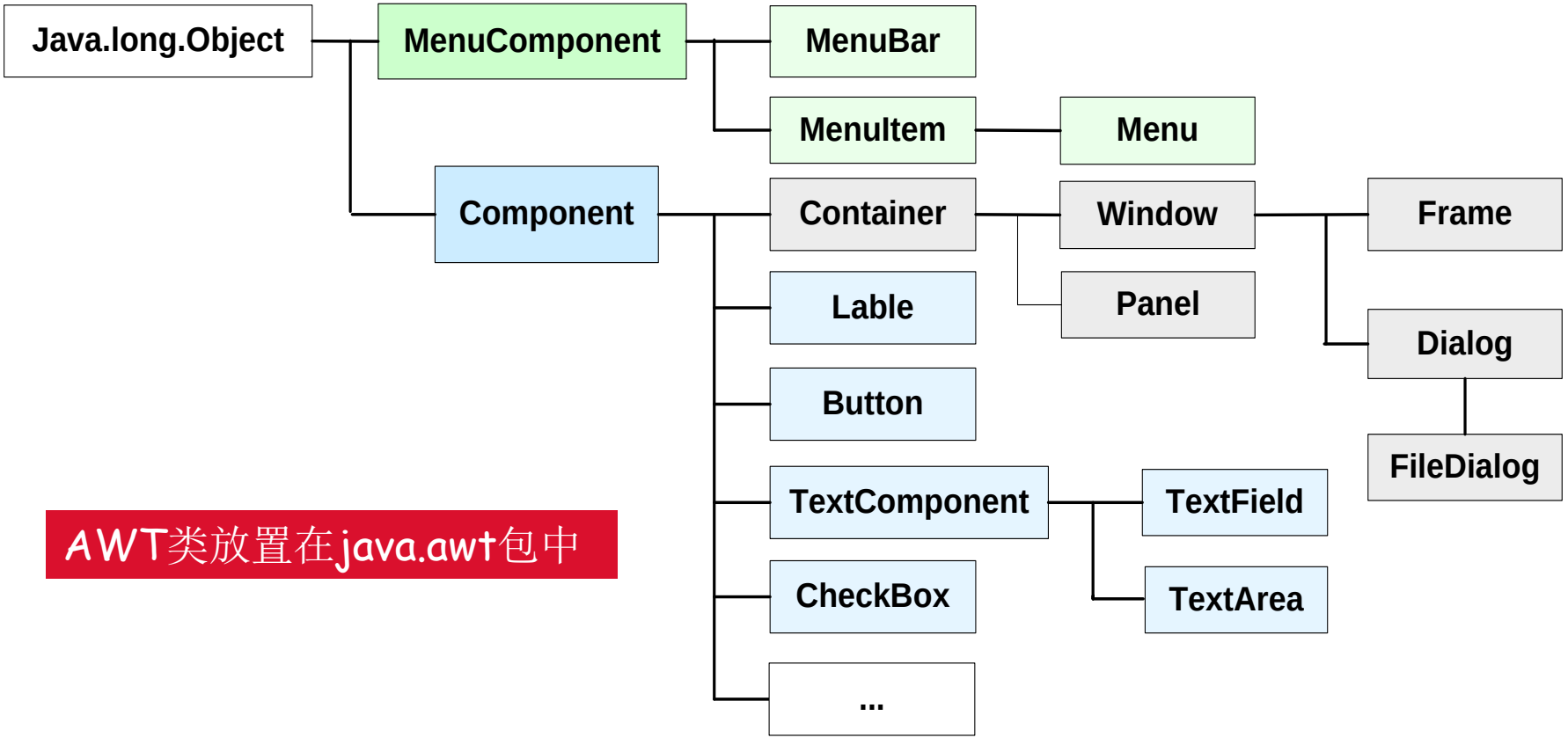
- 例 TestMultiFrame.java

在框架中加入其他组件

- 框架本身是一个容器。容器是图形用户界面中可以容纳其他组件的特殊组件，一个容器中可以容纳一个或多个组件，甚至还可以容纳其他容器。
- Container类是从Component类派生来的，从Component中派生的典型容器包括：Panel，Applet，ScrollPane，Window，Frame，Dialog和FileDialog。
  - 框架(**Frame**)
    - 构造方法: Frame(String title)
    - 一个Frame被创建后初始大小为(0,0)，需要设置大小
  - 面板(**Panel**)
    - 构造方法 Panel()
    - 面板必须放入Window或Frame中才可见。
    - 使用add()方法将Panel添加到Frame中。

AWT组件库

- 标签　按钮　单行文本　多行文本　单选框　复选框
- 菜单 下拉菜单
- 对话框　文件对话框



AWT类放置在java.awt包中

常用AWT组件

- **标签 (Label) - Java.awt.Label**
  - 在界面上显示一行文字。

    Label label = new Label(“Hello World!”)。

    void setText(String s);

    String getText();
  - 一般与事件无关
- **按钮 (Button)**

  Button b = new Button(“Help”);
  - 点击按钮会激发**ActionEvent**事件。
  - 使用**ActionListener**接口监听、处理该事件
- **单行文本 (TextField)**
  - 用来接受用户输入

    TextField t = new TextField(“name”, 10);
  - setEnable(false)设置为只读。
  - 在TextField处回车会激发ActionEvent事件
  - 使用ActionListener接口监听、处理该事件

常用AWT组件-2

- **多行文本 (TextArea)**
  - 用来接受用户输入的多行文本
  - TextArea t = new TextArea(“name”, 5，50);
  - setEnable(false)设置为只读
  - 在TextField处回车表示换行,不会激发ActionEvent事件
- **复选框 (Checkbox)**
  - 提供简单的on/off开关
  - Checkbox c1 = new Checkbox(“Red”,null, true);
  - setEnable(false)设置为只读
  - 当复选框发生改变时，会激发ItemEvent事件
- **单选框 (CheckboxGroup)**
  - 提供单项选择。每个选项是相关联、互斥的
  - 首先构造一个**CheckboxGroup**对象，然后为这个Group添加Checkbox
  - setEnable(false)设置为只读
  - 当复选框发生改变时，会激发ItemEvent事件

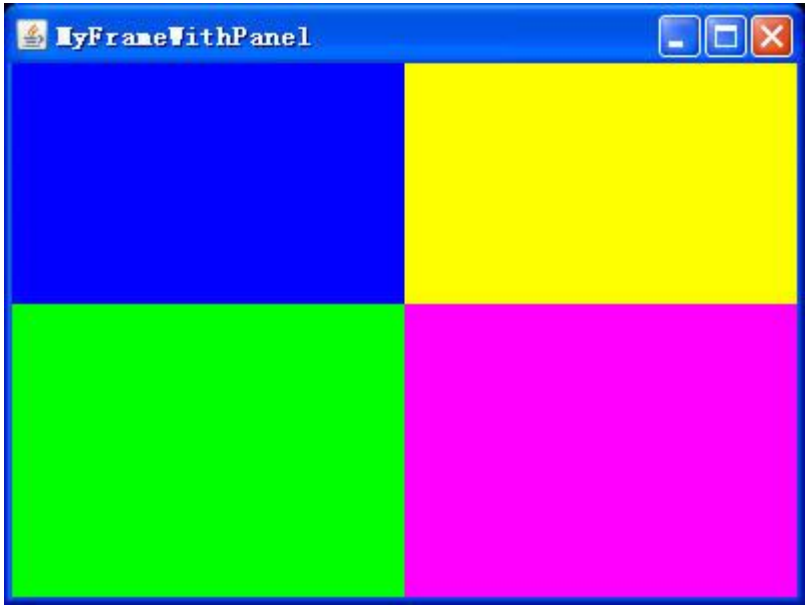
常用AWT组件-3

- **对话框 (Dialog)**
  - 可以实现用户数据的接收，提示一些信息等
  - 可分为模式和非模式对话框
  - Dialog d = new Dialog(parentFrame, “Title” , false);
  - setVisible(boolean)来控制对话框的可见性
- **文件对话框 (FileDialog)**
  - 用来为用户选择文件，是模式对话框
  - FileDialog fd = new FileDialog(parentFrame, “Title”);
  - setVisible(boolean)来控制对话框的可见性
  - String fileName = fd.getFile();
  - 一般与事件无关
- **下拉菜单 (Choice)**

GUI例 - Frame with Label

```
1.  //HelloGuiWorld Frame-Label
2.  import java.awt.*;
3.  import java.awt.event.*;
4.  class myFrame2 extends Frame {//新建的窗口类
5.          Label greetingLab;
6.          String helloStr;
7.          myFrame2(String s){
8.              super(s);
9.              this.setSize(300,200);
10.             this.setBackground(Color.yellow);
11.             //创建Label
12.           helloStr="图形版的Hellowrold!";
13.           greetingLab=new Label(helloStr);
14.           //将Label加到Frame
15.           this.add(greetingLab,"Center");
16.           this.setVisible(true);
17.       }
18. }
19. public class HelloGuiWorld02{
20.     public static void main(String[] args){
21.         myFrame2 frm=new myFrame2("HelloGUIWorld!");
22.     }
23. }
```

在框架中加入其他组件



- 例 TestMultiPanel.java

目　录

1 AWT 组件

2 布局管理器

3 Swing 包

界面布局管理器

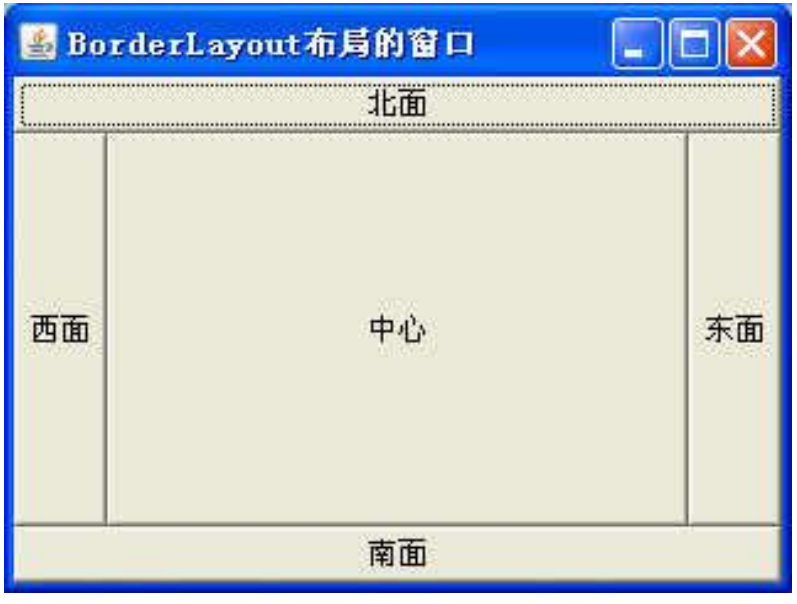
- 当把组件添加到容器中时，希望控制组件在容器中的位置。
- Java.awt包中共定义了5种布局类，每种布局都对应一种布局策略，分别是：
  - FlowLayout
  - BorderLayout
  - CardLayout
  - GridLayout
  - GridBagLayout
- 容器可以使用方法：　setLayout(布局对象);来设置自己的布局

FlowLayout

- FlowLayout的布局策略是将容器中的组件按照加入的先后顺序从左向右排列。如果一行排满转下一行继续，每行均采取居中排列。
- 是Panel型容器的默认布局，即Panel及其子类创建的容器对象，如果不专门为其指定布局，则它们的布局就是FlowLayout型布局。
- 例 FlowLayoutTest.java

BorderLayout

- BorderLayout布局策略是把容器内的空间划分为东、西、南、北、中5个区域分别用英文的East、West、South、North、Center表示，向容器中加入每个组件都要指明在容器的区域。
- 是Window型容器的默认布局，例如Frame、Dialog都是Window类的子类，它们的默认布局都是BorderLayout 布局。
- BorderLayoutTest.java



CardLayout

- 使用CardLayout的容器可以容纳多个组件，但是实际上同一时刻容器只能从这些组件中选出一个来显示，就像一叠“扑克牌”每次只能显示最上面一张一样，这个被显示的组件将占据所有的容器空间，依次排序。
- CardLayoutTest.java

GridLayout

- GridLayout的布局策略是把容器划分成若干行乘若干列的网格区域，组件就位于这些划分出来的小格中。
- GridLayout比较灵活，划分多少网格由程序自由控制，而且组件定位也比较精确.
- 由于GridLayout布局中每个网格都是相同大小并且强制组件与网格的大小相同，使得容器中的每个组件也都是相同的大小，显得很不自然。为了克服这个缺点，可以使用容器嵌套。
- GridLayoutTest.java

GridBagLayout

- 将组件排列在一行或一列，这取决于创建盒式布局对象时，是否指定了是行排列还是列排列。
- 使用行（列）型盒式布局的容器将组件排列在一行（列），组件按加入的先后顺序从左（上）向右（下）排列，容器的两端是剩余的空间。
- 和FlowLayout布局不同的是，使用行型盒式布局的容器只有一行（列），即使组件再多，也不会延伸到下一行（列），这些组件可能会被缩小大小，紧缩在这一行（列）中。

目　录

1 AWT 组件

2 布局管理器

3 Swing 包

swing包

- 为了克服AWT包中隐含的图形组件，在不同的操作平台上显示不同效果问题，1998年推出的JDK1.2版本时，将javax.swing包列入Java基础库。
- AWT与swing包的比较
  - swing 包拥有4倍于AWT的用户界面组件
  - 新版的swing包可能与旧版不兼容
  - swing包建立在AWT的基础上，不大可能完全舍弃AWT
  - AWT包的运行速度比swing包快
  - 将AWT包改写为swing包时，在很多情况下可以只在原有的AWT组件的每个类名前面加上“J”即可

详细内容见
软件包 javax.swing

创建JFrame窗口

JFrame类的构造方法	主要功能
JFrame()	创建没有标题的窗口
JFrame(String title)	创建以title 为标题的窗口
JFrame类的方法	**主要功能**
Container getContentPane()	获得窗口的ContentPane组件
int getDefaultCloseOperation()	当用户关闭窗口时的默认处理方法
int setDefaultCloseOperation()	设置用户关闭窗口时发生的操作
void update(Graphics g)	引用paint()方法重绘窗口
void remove(Component component)	将窗口中指定的组件删除
JMenuBar getMenuBar()	获得窗口中的菜单栏组件
void setLayout(LayoutManager manager)	设置窗口的布局

创建JFrame容器时注意

1、创建的JFrame 窗口是不可见的，要使得可见，需要使用show()方法或setVisible(boolean b)方法，其中setVisible中的参数b=true。

2、使用setSize方法设置窗口大小。

3、向JFrame 中添加组件时，必须先取得ContentPane，然后再使用add()方法把组件加入到ContentPane中，这与AWT包中的Frame直接使用add()方法添加组件不同。

- 将按钮组件加入到窗口的一般操作方法：

  1、设置窗口的ContentPane，创建按钮对象

  ```
  Container  con = getContentPane();
  btn=new JButton(“按钮1”);
  ```

  2、添加按钮组件到JFrame窗口中

  ```
  con.add(btn);
  ```

swing例子

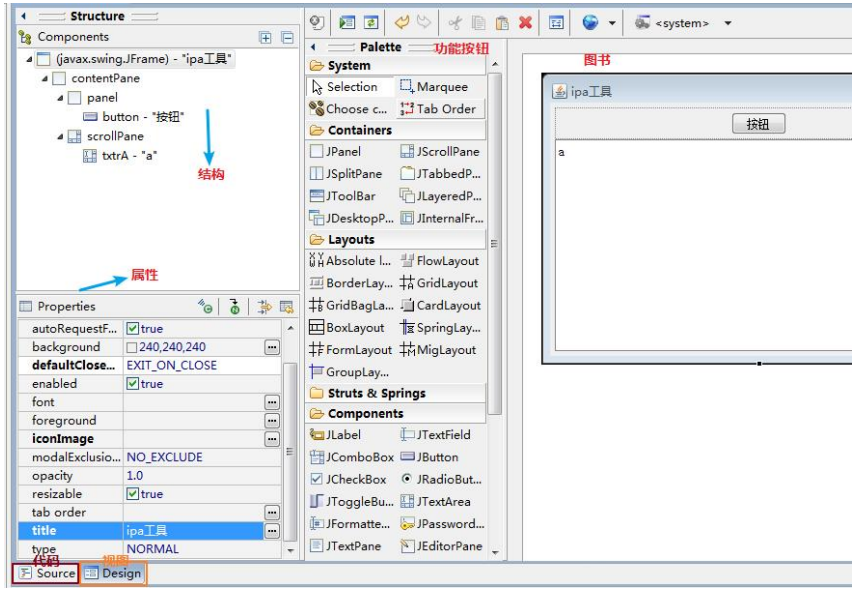
- 创建带有JLabel标签和JButton按钮的JFrame窗体。
  - JLabel标签中含有文字 和 图标Icon
  - 按下JButton时触发事件，弹出消息对话框
- jframetest.java

本章学习提示

- 本章我们了学习Java语言的用户界面的设计，及如何进行友好方便的人机交互。
  - Java中AWT包
  - 布局管理器的概念和使用
  - Java的事件处理机制
  - Swing包的简介

补充：

- 最方便的方法是用插件拖放制作，当然需要先了解完代码原理。
- Eclispe下常用windowbuilder。
  - https://www.eclipse.org/windowbuilder/



- IDEA 常用GUI插件：JFormDesigner
  - https://zhuanlan.zhihu.com/p/351621472

Thank You !