

Java SE程序设计 北京圣思园科技有限公司

主讲人 张龙

All Rights Reserved



构建图形用户界面（GUI）

- 课程目标

- - 描述AWT包及其组件
- - 定义Container、Component及Layout Manager等术语，以及它们是如何在一起来建立GUI的
- - 使用Layout Manager
- - 使用Flow、Border、Grid及Card布局管理器来获得期望的动态布局
- - 增加组件到Container
- - 正确运用Frame及Panel容器
- - 描述如何使用嵌套容器来完成复杂的布局



AWT概念

GUI全称是Graphical User Interface，即图形用户界面。

根据作用GUI组件可分为基本组件和容器。

组件又称构件，诸如按钮、文本框之类的图形界面元素

容器其实是一种比较特殊的组件，可以容纳其它组件，如窗口、对话框等

所有的容器类都是java.awt.Container的直接或间接子类。



AWT

- 提供基本的**GUI**组件，用在所有的Java applets及应用程序中
- 具有可以扩展的超类，它们的属性是继承的
- 确保显示在屏幕上的每个**GUI**组件都是抽象类组件的子类
- **Container**，它是**Component**的一个子类，而且包括两个主要子类
 - Panel
 - window

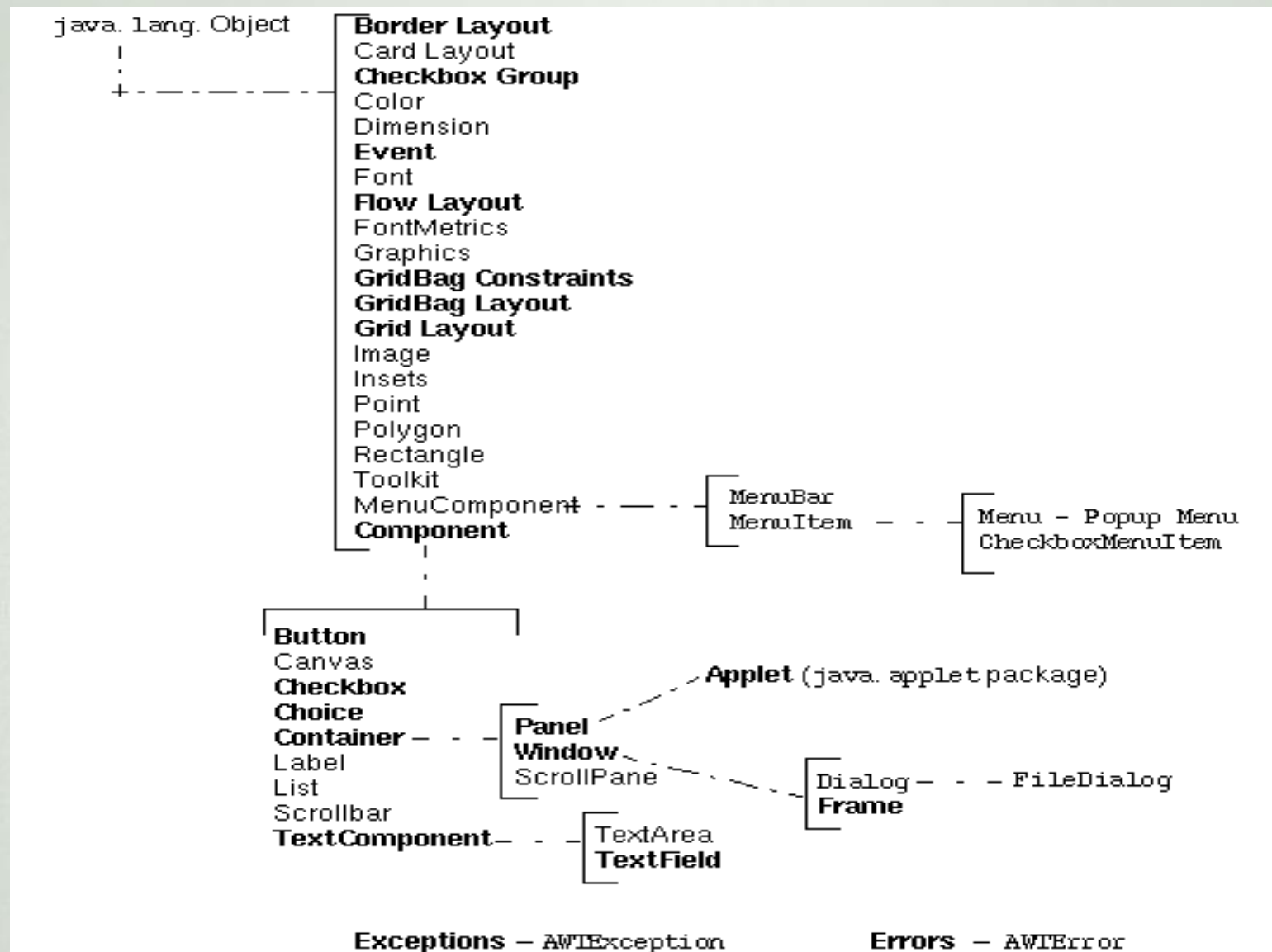


AWT

- AWT(Abstract Window Toolkit), 抽象窗口工具包, SUN公司提供的用于图形界面编程(GUI)的类库。基本的AWT库处理用户界面元素的方法是把这些元素的创建和行为委托给每个目标平台上(Windows、Unix、Macintosh等)的本地GUI工具进行处理。例如: 如果我们使用AWT在一个Java窗口中放置一个按钮, 那么实际上使用的是一个具有本地外观和感觉的按钮。这样, 从理论上来说, 我们所编写的图形界面程序能运行在任何平台上, 做到了图形界面程序的跨平台运行



Java.awt包



建立图形用户界面

- **Container**

- Container的两个主要类型是Window和Panel

- Window是Java.awt.Window的对象

- Window是java.awt.Window的对象。Window是显示屏上独立的本机窗口，它独立于其它容器
 - Window有两种形式：Frame(框架)和Dialog（对话框）。Frame和Dialog是Window的子类。Frame是一个带有标题和缩放角的窗口。对话框没有菜单条。尽管它能移动，但它不能缩放。



建立图形用户界面

- Panel是Java.awt.Panel的对象
 - Panel是Java.awt.Panel的对象。Panel包含在另一个容器中，或是在Web浏览器的窗口中。Panel确定一个四边形，其它组件可以放入其中。Panel必须放在Window之中（或Window的子类中）以便能显示出来
 - 注：容器不但能容纳组件，还能容纳其它容器，这一事实对于建立复杂的布局是关键性的，也是基本的。



定位组件

- 容器里的组件的位置和大小是由布局管理器决定的。
- 可以通过停用布局管理器来控制组件的大小或位置。
- 然后必须用组件上的`setLocation()`，`setSize()`，或`setBounds()`来定位它们在容器里的位置



定位组件

- 容器里的组件的位置和大小是由布局管理器决定的。容器对布局管理器的特定实例保持一个引用。当容器需要定位一个组件时，它将调用布局管理器来做。当决定一个组件的大小时，同样如此。布局管理器完全控制容器内的所有组件。它负责计算并定义上下文中对象在实际屏幕中所需的大小。



组件大小

- 因为布局管理器负责容器里的组件的位置和大小，因此不需要总是自己去设定组件的大小或位置。
- 如果必须控制组件的大小或位置，而使用标准布局管理器做不到，那就可能通过将下述方法调用发送到容器中来中止布局管理器：
 - `setLayout(null);`
- 做完这一步，必须对所有的组件使用 `setLocation()`, `setSize()` 或 `setBounds()`，来将它们定位在容器中。请注意，由于窗口系统和字体大小之间的不同，这种办法将导致从属于平台的布局。更好的途径是创建布局管理器的新子类



Frames

- 是Window的子类
- 具有标题和缩放角
- 从容器继承并以**add**方式添加组件
- 能以字符串规定的标题来创建**不可见**框架对象
- 能将BorderLayout当做缺省布局管理器
- 用**setLayout**方式来改变缺省布局管理器



Frames

- Frame是Window的一个子类。它是带有标题和缩放角的窗口。它继承于Java.awt.Container，因此，可以用add()方式来给框架添加组件。框架的缺省布局管理器就是Border Layout。它可以用setLayout()方式来改变



Frames

- 框架类中的构造程序 `Frame(String)` 用由 `String` 规定的标题来创建一个新的不可见的框架对象。当它还处于不可见状态时，将所有组件添加到框架中。

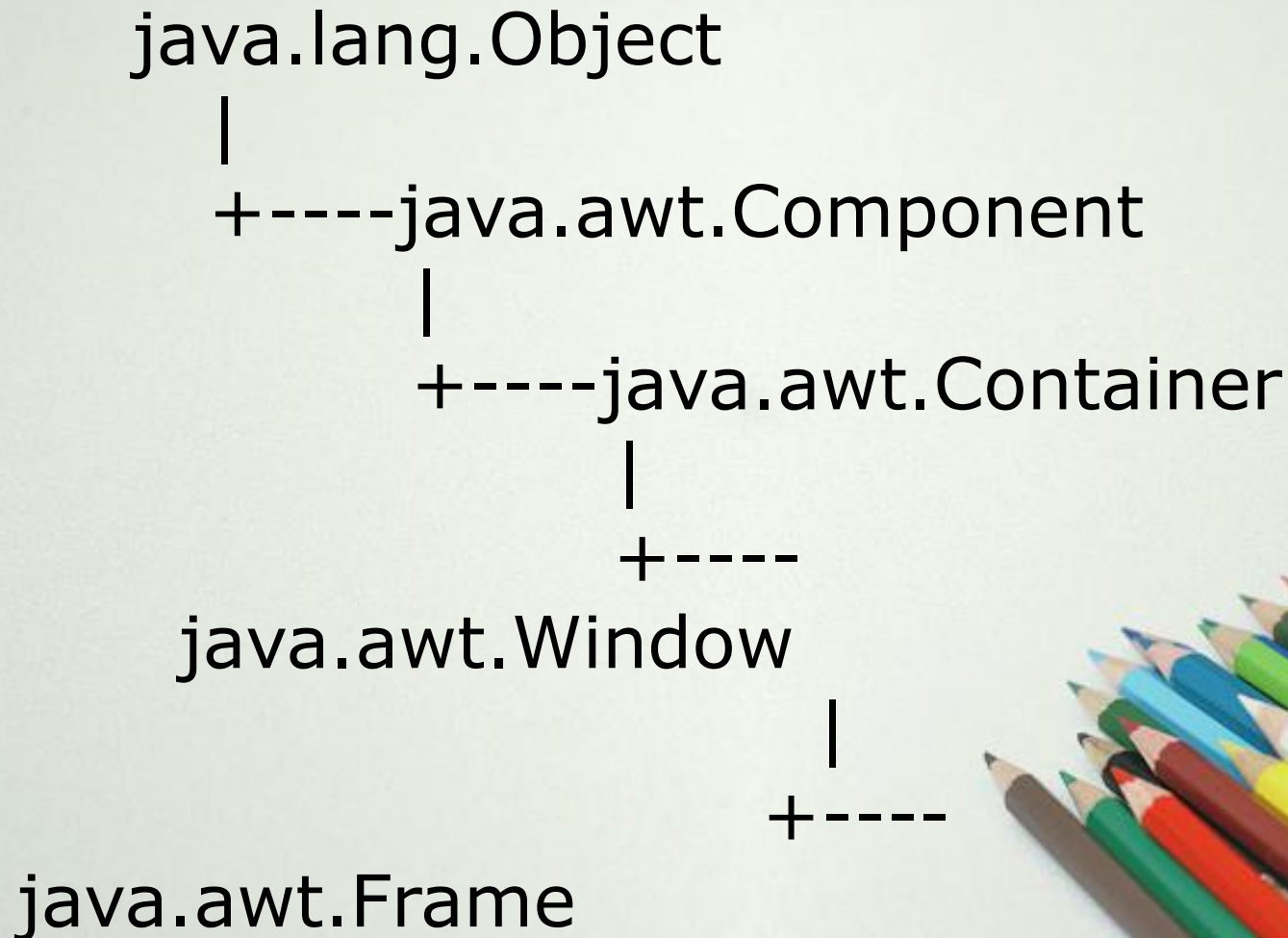


Frames

- 参见程序 **MyFrame.java**
 - 在框架显示在屏幕上之前，必须做成可见的（通过调用程序**setVisible(true)**），而且其大小是确定的（通过调用程序**setSize()**或**pack()**）



Frame类继承层次



Panels

- 为组件提供空间
- 允许子面板拥有自己的布局管理器
- 以**add**方法添加组件
 - 象**Frames**一样，**Panels**提供空间来连接任何**GUI**组件，包括其它面板。每个面板都可以有它自己的布管理程序。
 - 一旦一个面板对象被创建，为了能看得见，它必须添加到窗口或框架对象上。用**Container**类中的**add()**方式可以做到这一点



Panel类

- 提供容纳组件的空间
- 可以采用和所在容器不同的布局管理器
- Panel类的继承层次

java.lang.Object



+----java.awt.Component



+----java.awt.Container



+----java.awt.Panel

Panels

- 参见程序 `FrameWithPanel.java`
 - 程序创建了一个小的黄色面板，并将它加到一个框架对象上



布局管理器

- 容器里组件的位置和大小是由布局管理器来决定的。容器对布局管理器的特定实例保持一个引用。当容器需要定位一个组件时，它将调用布局管理器来完成。当决定一个组件的大小时，也是如此。
- 在AWT中，给我们提供了五种布局管理器：
 - **BorderLayout**
 - **FlowLayout**
 - **GridLayout**
 - **CardLayout**
 - **GridBagLayout**



布局管理器

- 容器中组件的布局通常由布局管理器控制。每个Container（比如一个Panel或一个Frame）都有一个与它相关的缺省布局管理器，它可以通过调用setLayout()来改变
- 布局管理器负责决定布局方针以及其容器的每一个子组件的大小。

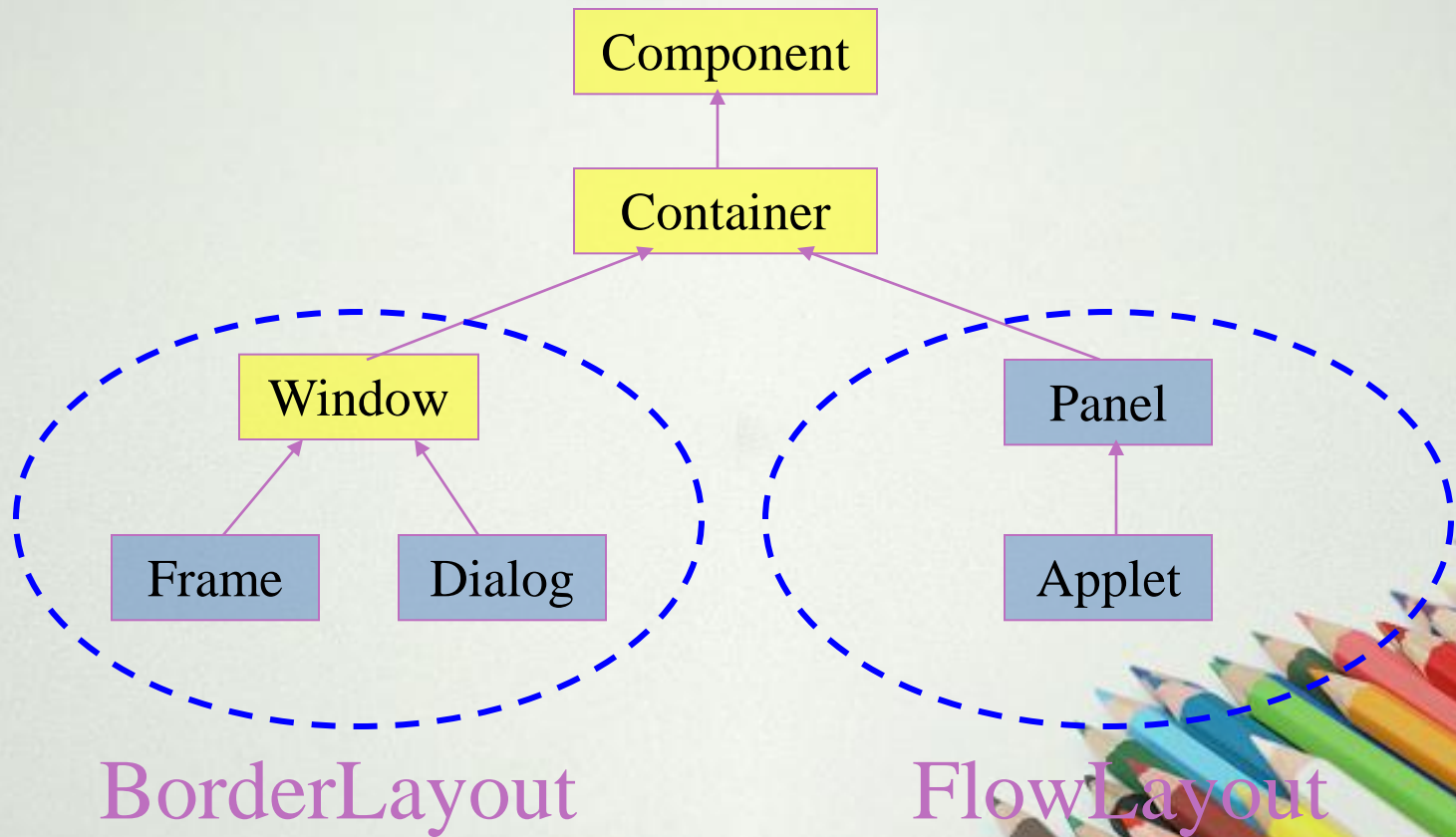


布局管理器

- 我们可以通过设置空布局管理器，来控制组件的大小和位置。调用**setLayout(null)**。
- 在设置空布局管理器后，必须对所有的组件调用**setLocation()**，**setSize()**或**setBounds()**，将它们定位在容器中。



默认布局管理器



布局管理器

- 参见程序 `ExGui.java`



Flow布局管理器

- 与其它布局管理器不一样，**Flow**布局管理器不限制它所管理的组件的大小，而是允许它们有自己的最佳大小。
- 默认是居中放置组件
- 如果想在组件之间创建一个更大的最小间隔，可以规定一个界限。
- 当用户对由**Flow**布局管理的区域进行缩放时，布局就发生变化



Flow布局管理器

- 下面的例子就是如何用类容器的setLayout()方法来创建Flow布局对象并安装它们
 - `setLayout(new FlowLayout(int align,int hgap, int vgap));`
 - 对齐的值必须是`FlowLayout.LEFT`, `FlowLayout.RIGHT`, 或 `FlowLayout.CENTER`
 - 例如:
 - `setLayout(new FlowLayout(FlowLayout.RIGHT, 20, 40));`



FlowLayout 的构造方法

- ✦ `new FlowLayout(FlowLayout.RIGHT, 20, 40);`
右对齐，组件之间水平间距20个像素，竖直间距40个像素；
- ✦ `new FlowLayout(FlowLayout.LEFT);`
左对齐，水平和竖直间距为缺省值：5；
- ✦ `new FlowLayout();`
使用缺省的居中对齐方式，水平和竖直间距为缺省值：5；



Flow布局管理器

- 参见程序 `MyFlow.java`



Border布局管理器

- Border布局管理器为在一个Panel或Window中放置组件提供一个更复杂的方案。Border布局管理器包括五个明显的区域：东、南、西、北、中。
- 北占据面板的上方，东占据面板的右侧，等等。中间区域是在东、南、西、北都填满后剩下的区域。当窗口垂直延伸时，东、西、中区域也延伸；而当窗口水平延伸时，东、西、中区域也延伸
- 当窗口缩放时，按钮相应的位置不变化，但其大小改变



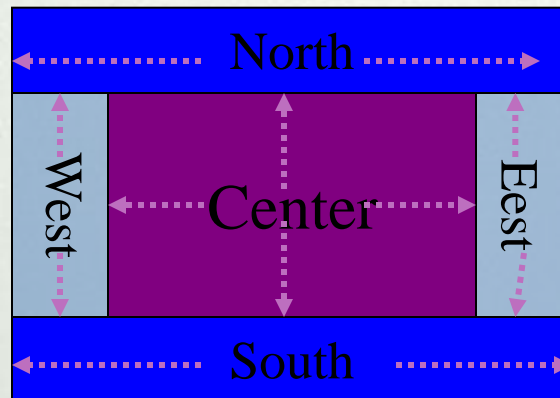
Border布局管理器

- BorderLayout是Frame类的默认布局管理器
- BorderLayout将整个容器的布局划分成东、西、南、北、中五个区域，组件只能被添加到指定的区域
- 如不指定组件的加入部位，则默认加入到Center区域
- 每个区域只能加入一个组件，如加入多个，则先前加入的组件会被遗弃



Border布局管理器

- BorderLayout型布局容器尺寸缩放原则
 - 北、南两个区域只能在水平方向缩放(宽度可调整)
 - 东、西两个区域只能在垂直方向缩放(高度可调整)
 - 中部可在两个方向上缩放



Border布局管理器

- 参见程序 ExGui2.java
 - 布局管理器给予南、北组件最佳高度，并强迫它们与容器一样宽。但对于东、西组件，给予最佳宽度，而高度受到限制



Grid布局管理器

- Grid布局管理器为放置组件提供了灵活性。用许多行和栏来创建管理程序。然后组件就填充到由管理程序规定的单元中。
- 比如，由语句`new GridLayout(3,2)`创建的有三行两栏的Grid布局能产生六个单元
- Grid布局管理器总是忽略组件的最佳大小。所有单元的宽度是相同的，是根据单元数对可用宽度进行平分而定的。同样地，所有单元的高度是相同的，是根据行数对可用高度进行平分而定的



Grid布局管理器

- 将组件添加到网格中的命令决定它们占有的单元。单元的行数是从左到右填充，就象文本一样，而列是从上到下由行填充。
- 参见程序 `GridEx.java`



Card布局管理器

- Card布局管理器能将界面看作一系列的卡，其中的一个在任何时候都可见。用add()方法来将卡添加到Card布局中。Card布局管理器的show()方法应请求转换到一个新卡中
- 参见程序 CardTest.java



GridBag布局管理器

- 除了Flow、Border、Grid和Card布局管理器外，核心Java. awt也提供GridBag布局管理器
- GridBag布局管理器在网格的基础上提供复杂的布局，但它允许单个组件在一个单元中而不是填满整个单元那样地占用它们的最佳大小。网格包布局管理器也允许单个组件扩展成不止一个单元
- 不作为重点



创建面板及复杂布局

- 参见程序 `ExGui3.java`
 - 是两个布局管理器的综合使用



布局管理器总结(1)

■ Frame

- Frame是一个顶级窗口。
- Frame的缺省布局管理器为BorderLayout。

■ Panel

- Panel无法单独显示，必须添加到某个容器中。
- Panel的缺省布局管理器为FlowLayout。
- 当把Panel作为一个组件添加到某个容器中后，该Panel仍然可以有自己的布局管理器。因此，可以利用Panel使得BorderLayout中某个区域显示多个组件。



布局管理器总结(2)

- 在程序中安排组件的位置和大小时，应注意：
 - 容器中的布局管理器负责各个组件的大小和位置，因此用户无法在这种情况下设置组件的这些属性。如果试图使用Java语言提供的`setLocation()`，`setSize()`，`setBounds()`等方法，则都会被布局管理器覆盖。
 - 如果用户确实需要亲自设置组件大小或位置，则应取消该容器的布局管理器，方法为：
`setLayout(null);`

