

Java AWT基础及布局管理

Java AWT基础及布局管理

GUI(Graphical User Interface) , 图形用户界面。

Java GUI分两种：

AWT(Abstract Window Toolkit)：抽象窗口工具集，是第一代的Java GUI组件。

绘制依赖于底层的操作系统。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。

Swing，不依赖于底层细节，是轻量级的组件。现在多是基于Swing来开发。

AWT概念

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。

GUI组件根据作用可以分为两种：基本组件和容器组件。

基本组件又称构件，诸如按钮、文本框之类的图形界面元素。

容器是一种比较特殊的组件，可以容纳其他组件，容器如窗口、对话框等。

所有的容器类都是**java.awt.Container**的直接或间接子类。

Container，它是Component的一个子类，而且包括**两个主要子类：Panel，Window**。

注：容器不但能容纳组件，还能容纳其他容器，这一事实对于建立复杂的布局是关键的，也是基本的。

容器Container

Container的两个主要类型是Window和Panel。

Window

Window是**java.awt.Window**的对象。

Window是显示屏上独立的本机窗口，它独立于其他容器。

Window有两种形式：Frame和Dialog，Frame和Dialog是Window类的子类。

Frame是一个带有标题和缩放角的窗口。

对话框有标题，没有菜单条，它能移动，但不能缩放。

Frame

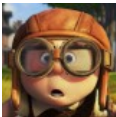
是Window的子类。从容器继承并以add的方式添加组件。

具有标题和缩放角。

能以字符串规定标题来创建Frame对象，刚创建出来的Frame对象是不可见的。

公告

欢迎来到圣骑士Wind的博客！



Dandan Meng

My Github

友情链接

我的github pages

WeYoung

yanni4night

微信公众号: 圣骑士Wind: 分享最新Android开发资讯



Links：

My Github

Android API Guides

GrepCode Android

AndroidXRef

昵称：圣骑士wind

园龄：5年10个月

粉丝：1266

关注：68

+加关注

<		2013年2月			
日	一	二	三	四	
27	28	29	30	31	
3	4	5	6	7	
10	11	12	13	14	
17	18	19	20	21	
24	25	26	27	28	
3	4	5	6	7	

搜索

当它还处于不可见状态时，将所有组件添加到框架中。调用setVisible(true)方法将Frame对象变为可见的。

将BorderLayout当做缺省布局管理器。

用setLayout方式来改变布局管理器。

田

Frame程序例子1

Panel

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，其他组件可以放入其中。

Panel必须放在Window或Window的子类之中，以便能显示出来。

为组件提供空间，允许子面板有自己的布局管理器。

以add方法添加组件。

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方法可以做到这一点。

定位组件

容器里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，或决定一个组件的大小时，它将调用布局管理器来做。

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。

可以通过**停用布局管理器(setLayout(null))**来控制组件的位置和大小。

然后就可以用组件上的 setLocation(), setSize(), setBounds()来定位它们在容器里的位置。

不过**建议使用布局管理器，而不是自己设定位置。**

布局管理器

在AWT中，提供了五种布局管理器：

BorderLayout

FlowLayout

GridLayout

CardLayout

GridBagLayout

布局管理器负责决定布局方针以及容器的每一个组件的大小。

默认的布局管理器：

找找看

谷歌搜索

常用链接

[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

[我的标签](#)

最新随笔

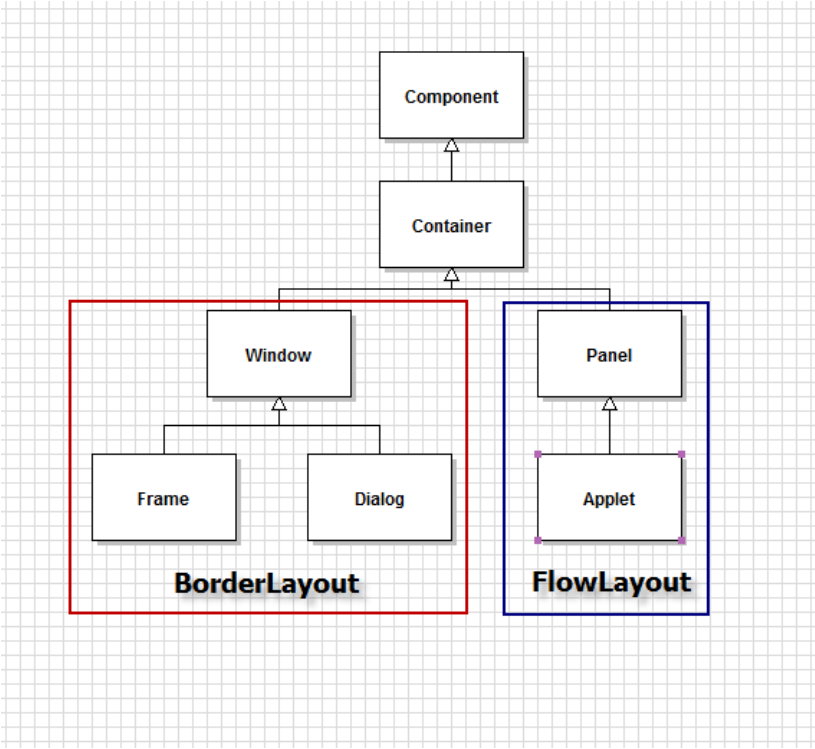
- 20+个很棒的Android开源项目
- 网络使用和电池消耗 原因和改进
- Android Realm数据库使用指南
- Say Hello to ConstraintLayout
- Android Weekly Notes Issue #237
- Android Weekly Notes Issue #236
- Android Weekly Notes Issue #235
- Android Weekly Notes Issue #234
- Android Weekly Notes Issue #233
- Effective Java笔记一 创建和销毁对象

我的标签

- Android(175)
- Java(86)
- Android Weekly(42)
- Web(38)
- 设计模式(29)
- JavaScript(29)
- RxJava(24)
- Kotlin(23)
- Animation(19)
- 数据结构和算法(17)
- 更多

随笔分类(507)

- 3D基础(3)
- Android(107)
- Android Animation(8)
- Android Testing(3)
- Android Weekly(43)
- Android 进阶(8)
- Android开发相关类库(3)
- C#(3)
- C++(1)
- CMake(2)
- DirectX(3)
- Git(7)
- GPU(1)
- Java(83)



JavaScript(24)
JNI(1)
jQuery(10)
JUnit(13)
JVM(7)
Linux(6)
MySQL(3)
OpenCV(2)
OpenGL(1)
OpenGL ES(1)
Python(2)
Testing(1)
Tools & IDE(2)
UX(5)
Web(28)
Windows(2)
XML(12)
XNA(1)
笔试面试(2)
编程调试(4)
服务器(3)
科研什么的(12)
每天一个算法题(10)
设计模式(29)
数据结构和算法(19)
数据库(7)
图形图像(2)
图形学(10)
网络(13)

随笔档案(429)

- 2017年6月 (1)
- 2017年5月 (4)
- 2017年4月 (6)
- 2017年3月 (5)
- 2017年2月 (6)
- 2017年1月 (4)
- 2016年12月 (5)
- 2016年11月 (6)
- 2016年10月 (8)
- 2016年9月 (5)
- 2016年8月 (4)
- 2016年7月 (1)
- 2016年6月 (5)
- 2016年5月 (1)
- 2016年3月 (1)
- 2015年10月 (1)
- 2015年6月 (1)
- 2015年5月 (3)
- 2015年4月 (5)
- 2015年3月 (1)
- 2015年2月 (1)
- 2015年1月 (6)
- 2014年12月 (4)
- 2014年9月 (2)
- 2014年7月 (3)
- 2014年6月 (1)
- 2014年5月 (4)
- 2014年4月 (17)
- 2014年3月 (10)
- 2014年2月 (11)
- 2014年1月 (3)
- 2013年12月 (4)
- 2013年11月 (7)
- 2013年10月 (2)

FlowLayout

与其他布局管理器不一样，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。

FlowLayout的构造方法：

- new FlowLayout();
使用缺省的居中对齐方式，水平和垂直间距为缺省值5。
- 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像素。

田

FlowLayout程序例子

BorderLayout

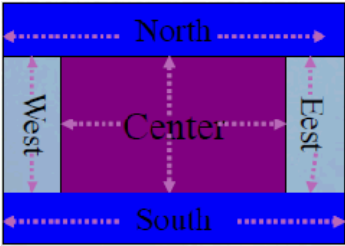
- BorderLayout为在一个Panel或Window中放置组件提供一个更复杂的方案。
- Border布局管理器包括五个明显的区域：东南西北中。
- 北占据面板的上方，东在右。中间区域是在东南西北都填满后剩下的区域。
- 没有指定方向的组件默认放在中间。
- 每个区域只能加入一个组件，如加入多个，则之前加入的组件会被遗弃。
- 当窗口垂直延伸时，东西中区域也延伸；而当窗口水平延伸时，南北中区域也延伸。
- 当窗口缩放时，按钮相应的位置不改变，但其大小改变。

缩放原则：

南北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。



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。

对于东西组件，给予最佳宽度，而高度受到限制。

BorderLayout程序例子

GridLayout

Grid布局管理器为放置组件提供了灵活性。用许多行和列来管理，组件就填充到由管理程序规定的单元中。

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。同样的，所有单元的高度也是相同的，是根据行数对高度进行平分而定的。

GridLayout是按照添加的顺序顺次填入单元，没有加够的，则后面的单元为空白。

CardLayout

Card布局管理器能将界面看做一系列的卡，其中的一个在任何时候都可见。

用add()方法将卡添加到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。

GridBagLayout

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。

网格包布局管理器也允许单个组件扩展成不止一个单元。

参考资料

张龙老师Java SE视频。

另附上本博客之前文章一篇：<http://www.cnblogs.com/mengdd/archive/2012/08/30/2663630.html>

分类: Java

标签: Java

好文要顶 关注我 收藏该文

圣骑士wind

关注 - 68 粉丝 - 1266

+加关注

10

- 2013年9月 (11)
- 2013年8月 (14)
- 2013年7月 (10)
- 2013年6月 (34)
- 2013年5月 (26)
- 2013年4月 (21)
- 2013年3月 (38)
- 2013年2月 (25)
- 2013年1月 (38)
- 2012年12月 (18)
- 2012年11月 (20)
- 2012年10月 (4)
- 2012年9月 (3)
- 2012年8月 (13)
- 2012年7月 (2)
- 2012年5月 (1)
- 2012年4月 (2)
- 2011年8月 (1)

文章档案(8)

- 2015年4月 (1)
- 2015年1月 (1)
- 2013年12月 (1)
- 2013年8月 (3)
- 2013年3月 (1)
- 2012年11月 (1)

Android学习

Android源码查看

Android源码项目git

老罗的Android之旅

官方文档

Android 官网

Git Reference

Python

Python官网

开源社区

GitHub

Google Code

学习链接

A Byte of Python

某G家程序员

积分与排名

积分 - 821586

排名 - 93

最新评论

- 1. Re:Java 多线程（四）多线程访问成员变量与局部变量这个局部变量和ThreadLocal有什么区别？

--mzzcy
- 2. Re:堆排序 Heap Sort
图给 满分

« 上一篇 : [Java 异常基础 Exception](#)

» 下一篇 : [Java 事件模型基础 : 监听器、事件、事件源、事件注册](#)

posted @ 2013-02-06 15:31 [圣骑士wind](#) 阅读(9883) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
- 【阿里云】云计算科技红利邀您免费体验！云服务器、云数据库等35+产品，6个月免费使用！
- 【免费】从零开始学编程，开发者专属实验平台免费实践！
- 【推荐】又拍云年中大促！现在注册，充值最高送4800元

- 13128 11933
- [流芒](#)
3. [Re:20+个很棒的Android开源项目](#)
[android](#)
- euler
4. [Re:Java 包装类 自动装箱和拆箱](#)
[打包带走](#) >.<
- BoomKe
5. [Re:Java 静态导入](#)
[收藏](#)
- Lumia1020

阅读排行榜

1. [Android PopupWindow的使用和分析\(209267\)](#)
2. [Java 数组基础\(126605\)](#)
3. [Android Fragment 基本介绍\(120390\)](#)
4. [Android Fragment和Activity\(114182\)](#)
5. [Android 抽屉效果的导航菜单实现\(113116\)](#)

评论排行榜

1. [Google Maps Android API V2的使用及问题解决\(151\)](#)
2. [OpenCV实现人脸检测\(14\)](#)
3. [Java 多线程（七）线程间的通信\(13\)](#)
4. [Android HTTP实例 使用GET方法和POST方法发送请求\(13\)](#)
5. [Android 抽屉效果的导航菜单实现\(13\)](#)

推荐排行榜

1. [Java 多线程（六）synchronized关键字详解\(19\)](#)
2. [Android Fragment 基本介绍\(15\)](#)
3. [Android PopupWindow的使用和分析\(13\)](#)
4. [Java 异常基础 Exception\(12\)](#)
5. [Android 抽屉效果的导航菜单实现\(12\)](#)