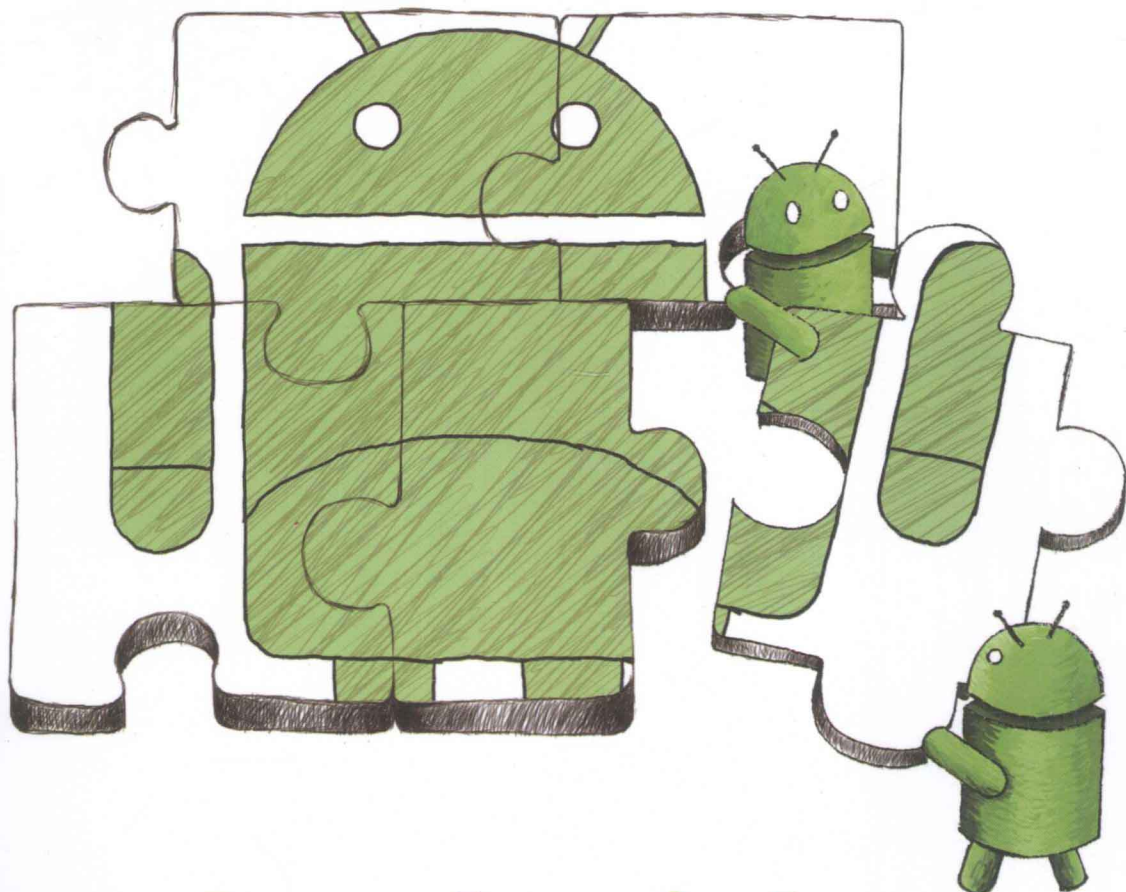


覆盖游戏、多媒体、地图、网络、平板等领域一百四十余个精彩实例

Broadview®
www.broadview.com.cn



Android

开发实例大全

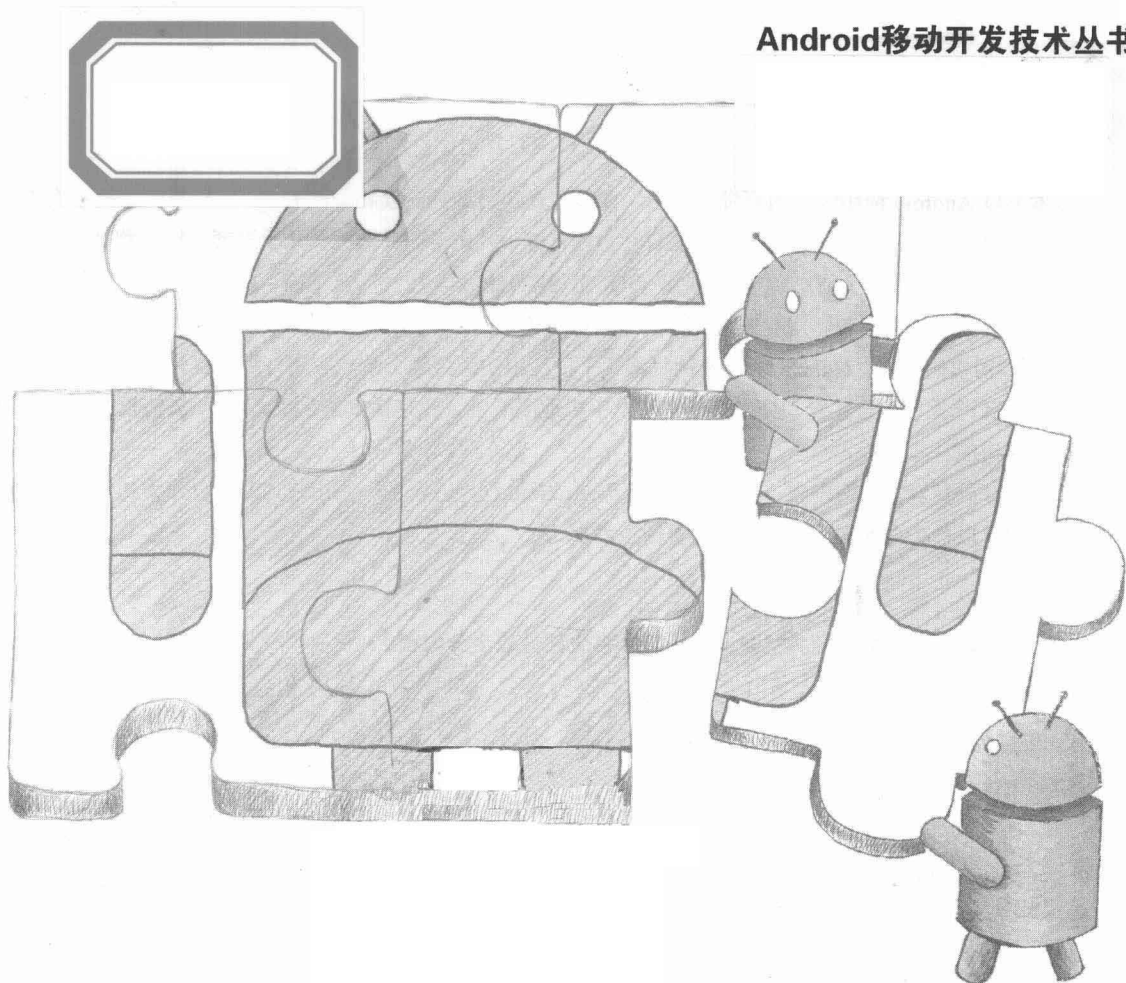
王东华 ▶ 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Android移动开发技术丛书



Android 开发实例大全

王东华 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以 Android 应用程序的开发为主题,并结合真实的案例向读者详细介绍了 Android 的基本组件的使用及应用程序开发的整个流程。本书的讲述由浅入深,实例全面并典型,几乎囊括了所有和 Android 应用相关的项目。全书分为 17 章,分别讲解了 UI 布局实例集锦、控件实例集锦、自动化服务实例集锦、数据存储实例集锦、电话和短信实例集锦、图形图像实例集锦、和网络有关的实例集锦、多媒体实例集锦、Google 地图实例集锦、Google API 实例集锦、平板应用实例集锦、游戏应用与程序优化。并且在本书最后,通过 4 个大型综合实例,分别介绍了开发手机地图系统、音乐播放器系统、NBA 激情投篮游戏、综合邮件系统的基本过程。

本书基础翔实,实例丰富,案例真实,适合 Android 程序开发的不同用户,既可以作为初学者的参考资料,也可以作为向此领域发展的程序员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Android 开发实例大全 / 王东华编著. —北京:电子工业出版社, 2012.8
(Android 移动开发技术丛书)
ISBN 978-7-121-17317-2

I. ①A… II. ①王… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2012)第 121355 号

策划编辑:张月萍

责任编辑:董 英

印 刷:北京中新伟业印刷有限公司

装 订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱

邮编:100036

开 本:787×1092 1/16

印张:45.75

字数:1171 千字

印 次:2012 年 8 月第 1 次印刷

印 数:3500 册 定价:89.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件到 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

随着 3G 的到来,无线带宽越来越宽,使得更多内容丰富的应用程序装入手机成为可能,如视频通话、视频点播、移动互联网冲浪和内容分享等。为了承载这些数据应用及快速部署,手机功能将会越来越智能,越来越开放。为了实现这些需求,必须有一个好的开发平台来支持,在此由 Google 公司发起的 OHA 联盟走在了业界的前列,2007 年 11 月推出了开放的 Android 平台,任何公司及个人都可以免费获取源代码及开发 SDK。由于其开放性和优异性,Android 平台得到了业界广泛的支持,其中包括各大手机厂商和著名的移动运营商等。继 2008 年 9 月第一款基于 Android 平台的手机 G1 发布之后,三星、摩托罗拉、索爱、LG 等主流手机制造商都推出了自己的 Android 平台手机。在 2011 年底,Android 超越了塞班和 iOS,雄踞智能手机市场占有率榜首的位置。

毕竟 Android 平台被推出的时间才短短 5 年,了解 Android 平台软件开发技术的程序员还不多,如何迅速地推广和普及 Android 平台软件开发技术,让越来越多的人参与到 Android 应用的开发中,是整个产业链都在关注的一个话题。为了帮助开发者更快地进入 Android 开发行列,笔者特意精心编写了本书。本书以 Android 应用程序的开发为主题,并结合真实的案例向读者详细介绍了 Android 的基本组件的使用及应用程序开发的整个流程。本书的讲述由浅入深,实例全面并典型,几乎囊括了所有和 Android 应用相关的项目。

从技术角度而言,Android 是一种融入了全部 Web 应用的平台。随着版本的更新,从最初的触屏到现在的多点触摸,从普通的联系人到现在的数据同步,从简单的 Google Map 到现在的导航系统,从基本的网页浏览到现在的 HTML 5,这都说明 Android 已经逐渐稳定,而且功能越来越强大。此外,Android 平台不仅支持 Java、C、C++等主流的编程语言,还支持 Ruby、Python 等脚本语言,Google 甚至专为 Android 应用开发推出了 Simple 语言,这使得 Android 有着非常广泛的开发群体。



本书的内容

在本书的内容中,按不同的门类提供了 138 个实例,并在本书的最后提供了 4 大完整案例,主要内容如下。

章	主要内容
第1章	UI布局实例
第2章	控件开发实例
第3章	自动化服务实例



章	主要内容
第4章	数据存储实例
第5章	电话和短信实例
第6章	图形图像实例
第7章	网络实例
第8章	多媒体实例
第9章	Google地图实例
第10章	Google API实例
第11章	平板实例
第12章	游戏实例
第13章	Android程序优化
第14章	手机地图系统
第15章	音乐播放系统
第16章	NBA激情投篮
第17章	邮件系统

参与本书编写的人员有：王东华、王振丽、熊斌、朱桂英、周秀、邓才兵、罗红仙、王石磊、孙宇、程娟、王文忠、王梦、陈强、于洋、管西京。本团队由于时间和水平所限，书中难免有不足之处。如有纰漏和不尽如人意之处，诚请读者提出意见或建议，以便修订并使之更臻完善。另外，为了更好地为读者服务，我们专门提供了技术支持网站 www.topchuban.com，欢迎读者光临论坛，无论是书中的疑问，还是学习过程中的疑惑，本团队将尽力为大家解答。请访问 www.broadview.com.cn/17317 下载本书的相关资源。

编 者
2012 年 7 月

目 录

第 1 章 UI 布局开发实例集锦..... 1	实例 022: 实现 QQ 空间中的显示照片效果..... 40
实例 001: 使用线性布局 (LinearLayout) 来布局屏幕..... 1	第 2 章 控件开发实例集锦..... 44
实例 002: 使用相对布局 (RelativeLayout) 来布局屏幕..... 3	实例 023: 显示在文本框中输入的信息..... 44
实例 003: 使用表格布局 (TableLayout) 来布局屏幕..... 5	实例 024: 使用 RadioGroup 控件实现选项选择功能..... 46
实例 004: 使用绝对布局 (AbsoluteLayout) 来布局屏幕..... 6	实例 025: 使用 CheckBox 控件实现同意条款效果..... 48
实例 005: 使用标签布局 (TabLayout) 来布局屏幕..... 8	实例 026: 使用 Spinner 控件实现选项选择... 51
实例 006: 使用层布局 (FrameLayout)..... 10	实例 027: 使用 Gallery 控件实现个人相簿功能..... 55
实例 007: 创建一个桌面组件 Widget..... 11	实例 028: 开发一个文件搜索程序..... 58
实例 008: 在屏幕中实现一个按钮效果..... 12	实例 029: 单击按钮后实现按钮置换..... 60
实例 009: 在屏幕中显示文字..... 13	实例 030: 模拟实现一个时钟界面效果..... 62
实例 010: 在屏幕中显示编辑框..... 17	实例 031: 联合使用 DatePicker 和 TimePicker 来选择日期和时间..... 65
实例 011: 在屏幕中显示复选框..... 18	实例 032: 动态分隔手机屏幕..... 68
实例 012: 在屏幕中显示单选框..... 21	实例 033: 使用 ListActivity 实现动态菜单列表..... 71
实例 013: 在屏幕中显示下拉列表框..... 22	实例 034: 加载手机磁盘中的文件..... 73
实例 014: 在屏幕中实现自动输入文本效果..... 26	实例 035: 实现抽屉样式效果..... 76
实例 015: 使用日期选择器控件 DatePicker..... 28	实例 036: 在屏幕中动态显示多种对话框... 82
实例 016: 自动选择一个输入时间..... 30	实例 037: 手机秘书——重要事情提醒..... 88
实例 017: 实现屏幕自动滚动..... 32	实例 038: 动态添加或删除菜单..... 96
实例 018: 实现一个进度条效果..... 32	实例 039: 自定义 Android 控件实现单选按钮..... 100
实例 019: 开发一个评分程序..... 35	
实例 020: 在屏幕中显示一幅指定的图片... 36	
实例 021: 设置一幅图片作为按钮来使用... 38	



实例 040: 使用自定义控件在屏幕中绘制一条虚线.....	104
实例 041: 在屏幕中实现手风琴效果	106
第 3 章 自动化服务实例集锦	109
实例 042: 自动获取手机屏幕的分辨率	109
实例 043: 自动获取手机电池的剩余电量	112
实例 044: 来电时自动发送信息提醒	115
实例 045: 获取手机存储卡的容量	119
实例 046: 操作存储卡和内存卡中的数据	122
实例 047: 设置闹钟	129
实例 048: 实现黑名单来电自动静音处理	135
实例 049: 设置一个自动开机界面	138
实例 050: 更换手机屏幕背景图片系统	140
实例 051: 设置自动更换手机桌面背景	144
实例 052: 自动控制后台系统服务	154
实例 053: 在屏幕中拖动一个按钮	157
第 4 章 数据存储实例集锦	163
实例 054: 在屏幕中显示 SharedPreferences 中存储的信息.....	163
实例 055: 演示数据添加、删除等操作	165
实例 056: 编写一个手机日记本程序	170
实例 057: 使用 ContentProvider 开发一个手机日记本.....	177
实例 058: 使用文件保存数据.....	188
实例 059: 将网上的图片保存到 SD 卡并在手机中显示出来.....	190
实例 060: 保存联系人信息.....	194
第 5 章 电话和短信实例集锦	197
实例 061: 编写一个拨号程序.....	197
实例 062: 发送一条短信.....	200
实例 063: 单击按钮实现拨号功能.....	203
实例 064: 一个通用发短信程序	206
实例 065: 收到短信后用 E-mail 邮件进行通知.....	210
实例 066: 实现触屏拨号功能.....	214
实例 067: 一个短信群发程序.....	215
实例 068: 来电后自动发送邮件通知.....	219
实例 069: 监听手机拨接状态.....	220
实例 070: 检查短信是否发送成功.....	223
第 6 章 图形图像实例集锦.....	229
实例 071: 在屏幕中绘制一个矩形.....	229
实例 072: 在屏幕中绘制一个画布.....	232
实例 073: 实现水纹的透明效果.....	235
实例 074: 渲染几何图形	239
实例 075: 用图片为背景实现动画效果.....	242
实例 076: 在手机屏幕中实现 Frame 动画效果.....	244
实例 077: 对图片进行缩放处理.....	246
实例 078: 旋转屏幕中的图片	250
实例 079: 实现旗帜飘扬的效果.....	253
实例 080: 模拟构建一个 3D 场景效果.....	258
实例 081: 模拟实现粒子系统效果.....	261
实例 082: 在 Canvas 中写文字	265
实例 083: 移动屏幕中的图片	272
实例 084: 在屏幕中绘制一个三维圆柱体	276
实例 085: 获取并显示图片的宽和高.....	285
实例 086: 编写一个手机屏保程序.....	287
实例 087: 以触摸方式移动图片	295
第 7 章 和网络有关的实例集锦	299
实例 088: 传递 HTTP 参数	299
实例 089: 在屏幕中使用 HTML 程序	303
实例 090: 在手机上浏览网页	304
实例 091: 使用内置浏览器打开网页.....	305

实例 092: 显示 QQ 空间中的照片.....	307	第 10 章 Google API 实例集锦.....	446
实例 093: 通过 RSS 系统在线浏览新闻 ...	309	实例 120: 模拟验证官方账号.....	446
实例 094: 下载网络图片作为手机背景.....	320	实例 121: 在手机中开发一个 Google 搜索 程序.....	453
实例 095: 将文件上传至服务器.....	323	实例 122: 在手机中编写一个翻译软件	458
实例 096: 开发一个移动博客发布者.....	326	实例 123: 生成一个二维码.....	460
实例 097: 远程下载并安装一个软件.....	331	实例 124: 使用 Google Chart API 生成二 维码.....	463
实例 098: 控制蓝牙系统.....	336		
第 8 章 多媒体实例集锦.....	346	第 11 章 平板应用.....	469
实例 099: 播放一个指定的 MP3 文件.....	346	实例 125: 开发一个音乐播放器.....	469
实例 100: 调节手机音量的大小.....	350	实例 126: 开发一个在线天气播报系统	475
实例 101: 实现相机预览和拍照功能.....	353	实例 127: 开发一个网页浏览程序.....	485
实例 102: 在手机中播放影片.....	360	实例 128: 使用文件来保存音乐状态.....	491
实例 103: 手机录音系统.....	363	实例 129: 在屏幕中显示不同的进度条 效果.....	494
实例 104: 设置手机中的铃声.....	368		
实例 105: 在线播放网络中的 MP3.....	371	第 12 章 游戏应用.....	498
实例 106: 在线下载音乐作为手机铃声.....	379	实例 130: 一个简单的贪吃蛇游戏.....	498
实例 107: 播放 GIF 格式的动画.....	385	实例 131: 开发一个魔塔游戏.....	504
实例 108: 在手机中播放 MP4 视频.....	391	实例 132: 疯狂足球游戏.....	524
实例 109: 在线观看 3GP 视频.....	394		
第 9 章 Google 地图实例集锦.....	403	第 13 章 Android 程序优化.....	547
实例 110: 定位当前位置的坐标.....	403	实例 133: 测试计算机的性能.....	547
实例 111: 在手机中使用 Google 地图.....	406	实例 134: 测试内存性能.....	550
实例 112: 在地图中定位输入的坐标.....	413	实例 135: 优化 Android 模拟器.....	553
实例 113: 在地图中快速查询某个位置.....	416	实例 136: 使用 merge 标签优化 UI 界面....	556
实例 114: 随着手机的移动自动更新位置	418	实例 137: 优化 Android Layout.....	559
实例 115: 在手机地图中描绘线路并计算 线路距离.....	423	实例 138: 优化 Bitmap 图片.....	560
实例 116: 在 Google 地图中显示指定的 位置.....	430	第 14 章 综合实例——手机地图系统.....	563
实例 117: 在手机地图中查询一个地址.....	432	实例 139: 使用 Google 地图开发一个综合 地图系统.....	563
实例 118: 在手机中实现地址查询.....	434	14.1 项目分析.....	563
实例 119: 开发一个路径导航系统.....	438	14.1.1 规划 UI 界面.....	563
		14.1.2 数据存储设计.....	564



14.2	具体实现	565
14.2.1	新建工程	565
14.2.2	主界面	565
14.2.3	新建界面	567
14.2.4	设置界面	570
14.2.5	帮助界面	574
14.2.6	地图界面	576
14.2.7	数据存取	587
14.2.8	实现 Service 服务	592
14.3	发布自己的作品来赢利	594
14.3.1	申请成为会员	594
14.3.2	生成签名文件	597
14.3.3	使用签名文件	602
14.3.4	发布	604

第 15 章 综合实例——音乐播放器系统.....605

实例 140:	开发一个音乐播放器	605
15.1	项目介绍	605
15.1.1	项目背景介绍	605
15.1.2	项目目的	606
15.1.3	对读者的意义	606
15.2	需求分析	606
15.2.1	构成模块	606
15.2.2	功能结构图	609
15.2.3	系统功能说明	610
15.2.4	总体目标	610
15.3	设计数据库	611
15.3.1	字段设计	611
15.3.2	数据库连接	611
15.3.3	创建数据库	612
15.3.4	操作数据库	613
15.3.5	数据显示	614
15.4	具体编码	615
15.4.1	服务信息主界面	615
15.4.2	播放器主界面	616
15.4.3	播放列表功能	628
15.4.4	菜单功能模块	631
15.4.5	播放设置界面	634
15.4.6	设置显示歌词	638
15.4.7	文件浏览器模块	639

第 16 章 体育类游戏——NBA 激情投篮644

实例 141:	NBA 激情投篮	644
16.1	项目介绍	644
16.1.1	意义和技术分析	644
16.1.2	策划游戏	645
16.1.3	准备工作	645
16.2	项目架构	646
16.2.1	总体架构	646
16.2.2	规划类	646
16.3	具体编码	648
16.3.1	编写主类 LanqiuActivity	648
16.3.2	编写常量类	652
16.3.3	设计游戏试玩界面	658
16.3.4	绘制游戏场景	663

第 17 章 综合邮件系统672

实例 142:	开发一个综合邮件系统	672
17.1	项目介绍	672
17.1.1	项目背景介绍	672
17.1.2	项目目的	672
17.2	系统需求分析	673
17.2.1	构成模块	673
17.2.2	系统流程	675
17.2.3	功能结构图	676
17.2.4	系统功能说明	677
17.2.5	系统需求	677
17.3	数据存储设计	677
17.3.1	用户信息类	678
17.3.2	SharedPreferences	682
17.4	具体编码	684
17.4.1	欢迎界面	684
17.4.2	系统主界面	687
17.4.3	邮箱类型设置	694
17.4.4	邮箱收取设置	697
17.4.5	邮箱发送设置	703
17.4.6	邮箱用户检查	708
17.4.7	设置用户别名	713
17.4.8	用户邮件编辑	716
17.5	总结	723

第 1 章 UI 布局开发实例集锦

外观向来是工业产品的设计核心，是激发用户购买欲望的主要因素之一。本章讲的 UI 布局指的是手机界面布局，一款手机的屏幕界面效果是吸引用户购买的重要元素之一，因为消费者更倾向于选择界面美观的产品。在设计优美的界面之前，一定要先对屏幕进行布局。本章将使用具体实例的实现过程介绍在 Android 系统中规划 UI 界面的方法。



实例 001：使用线性布局（LinearLayout）来布局屏幕

源码路径：daima\001

知识点介绍

在 Android 布局中，需要了解视图容器组件——ViewGroup 的概念，使用视图容器组件 ViewGroup 的语法格式如下：

```
android.view.Viewgroup
```

ViewGroup 的功能是包含并管理下级系列的 Views 和其他 ViewGroup，是一个布局的基类。类 ViewGroup 好像一个 View 容器，负责对添加进来的 View 进行布局处理。一个 ViewGroup 可以添加到另一个 ViewGroup 中去。这是因为 ViewGroup 也继承于 View.Viewgroup 类，是其他容器类的基类。它们之间的关系如图 1-1 所示。

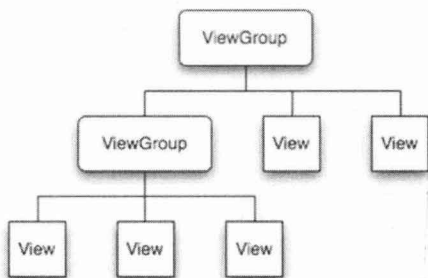


图 1-1 各个类的继承关系

我们知道，一个 Android 程序是由一个或多个 Activity 组成的，每个 Activity 是一个 UI 容器，Activity 本身不在用户界面中显示出来。在 Android 中，类 View 起了一个非常重要的作用，View 是一个最基本的 UI 类，几乎所有的 UI 组件都是继承于 View 而实现的。



使用 View 的语法格式如下所示。

```
android.view.View
```

线性布局即 `LinearLayout` 布局，是 Android 屏幕中常用的布局方式之一，功能是垂直地或水平地显示 `ViewGroup` 的子视图（`View`）元素。

具体实现

step 1 使用 Eclipse 创建一个名为“001”的 Android 工程。

step 2 编写布局文件“`res/layout/main.xml`”，代码如下所示。

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal">
    <Button android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第一个按钮"
        android:layout_weight="1"
        />
    <Button android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第二个按钮"
        android:layout_weight="1"
        />
    <Button android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第三个按钮"
        android:layout_weight="1"
        />
    <Button android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第四个按钮"
        android:layout_weight="1"
        />
    <Button android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第五个按钮"
        android:layout_weight="1"
        />
</LinearLayout>
```

在上述代码中，在根 `LinearLayout` 视图组（`ViewGroup`）中包含了 5 个按钮（`Button`），它的子元素是以线性方式水平布局的。上述代码的运行效果如图 1-2 所示。

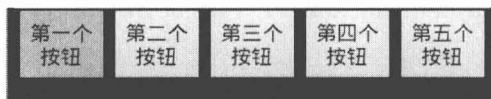


图 1-2 `LinearLayout` 布局效果



实例 002：使用相对布局（`RelativeLayout`）来布局屏幕

源码路径：daima\002

知识点介绍

相对布局是设计用户界面的有力工具，因为它消除了嵌套视图组。相对布局是指一个 `ViewGroup` 以相对位置显示它的子视图（`View`）元素，一个视图可以指定相对于它的兄弟视图的位置（如在给定视图的左边或者下面）或相对于 `RelativeLayout` 的特定区域的位置（如底部对齐或中间偏左）。

`RelativeLayout` 允许子元素指定它们相对于其他元素或父元素的位置（通过 ID 指定）。所以可以用右对齐、上下对齐或置于屏幕中央的形式来排列两个元素。如果第一个元素在屏幕的中央，那么相对于这个元素的其他元素将以屏幕中央的相对位置来排列。如果使用 XML 来指定这个布局，在定义它之前，被关联的元素必须定义。`RelativeLayout` 结构如图 1-3 所示。

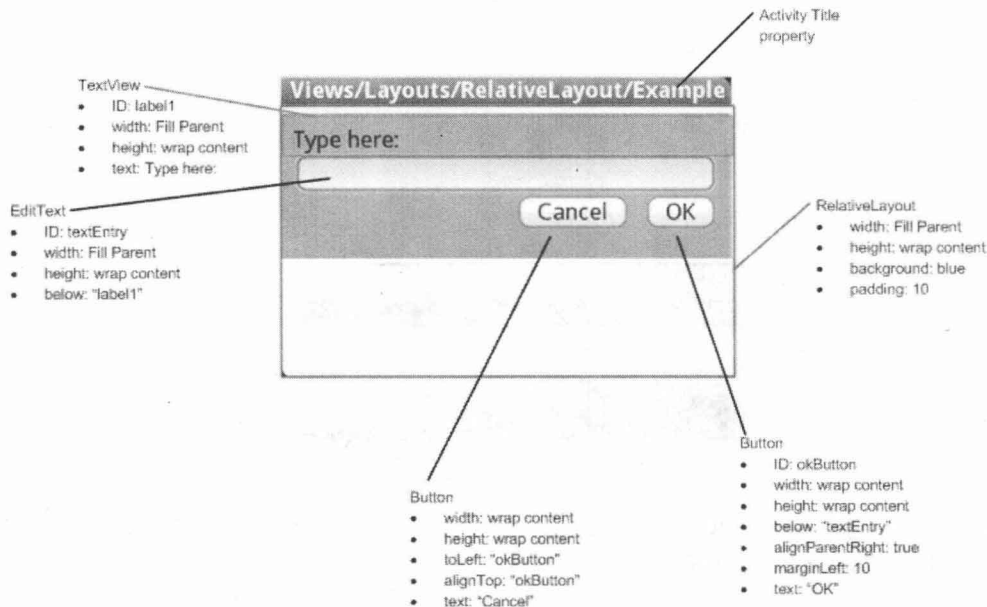


图 1-3 `RelativeLayout` 结构



具体实现

- step 1** 使用 Eclipse 创建一个名为“002”的 Android 工程。
- step 2** 编写布局文件“res/layout/main.xml”，代码如下所示。

```
<TextView
    android:id="@+id/label"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="请输入信息:" />
<EditText
    android:id="@+id/entry"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="@android:drawable/editbox_background"
    android:layout_below="@id/label" />
<Button
    android:id="@+id/ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/entry"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="10dip"
    android:text="确定" />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toLeftOf="@id/ok"
    android:layout_alignTop="@id/ok"
    android:text="取消" />
```

执行后的效果如图 1-4 所示。

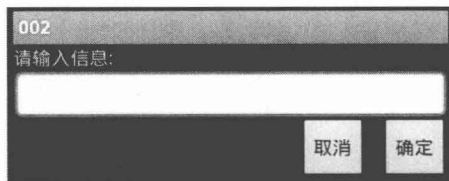


图 1-4 执行效果

在上述实例中，在 RelativeLayout 视图组中包含了一个 TextView、一个 EditText 和两个 Button，并且使用了下面的属性代码来设置布局的显示。

```
android:layout_below="@id/label"/>
android:layout_below="@id/entry"
android:layout_alignParentRight="true"
```

```
android:layout_toLeftOf="@id/ok"
android:layout_alignTop="@id/ok"
```

在使用相对布局方式时，正是使用这些类似的属性来定位视图到你想要的位置的，它们的值是你参照的视图的 ID。这些属性的意思很简单，就是英文单词的直译，这里就不多做介绍了。



实例 003：使用表格布局（TableLayout）来布局屏幕

源码路径：daima\003

知识点介绍

表格布局（TableLayout）其实是一个 ViewGroup 以表格显示它的子视图（View）元素，即行和列标识一个视图的位置。其实 Android 的表格布局与 HTML 中的表格布局非常类似，TableRow 就像 HTML 表格的<tr>标记。表格布局通常用于把子元素放入行与列中，不显示行、列或单元格边界线，但是单元格不能横跨行，像 HTML 中一样。表格布局效果如图 1-5 所示。

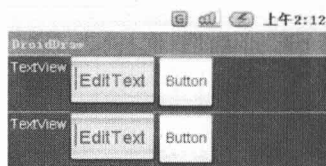


图 1-5 表格布局效果

在使用表格布局时需要了解如下 3 点。

- android:shrinkColumns: 对应的方法是 setShrinkAllColumns(boolean)，作用是设置表格的列是否收缩（列编号从 0 开始，下同），如果有多列则用逗号隔开（下同），如 android:shrinkColumns="0,1,2"，表示表格的第 1、2、3 列的内容是收缩的以适合屏幕，不会挤出屏幕。
- android:collapseColumns: 对应的方法是 setColumnCollapsed(int,boolean)，作用是设置表格的列是否隐藏。
- android:stretchColumns: 对应的方法是 setStretchAllColumns(boolean)，作用是设置表格的列是否拉伸。

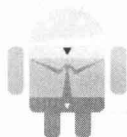
具体实现

step 1 使用 Eclipse 创建一个名为“003”的 Android 工程。

step 2 编写布局文件“res/layout/main.xml”，代码如下所示。

```
<TableRow>
<Button android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第一个按钮"
        android:layout_column="0"
        />

<Button android:id="@+id/button2"
        android:layout_width="wrap_content"
```



```
        android:layout_height="wrap_content"
        android:text="第二个按钮"
        android:layout_column="1"
    />
</TableRow>
<TableRow><!-- row2 -->
<Button android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第三个按钮"
        android:layout_column="1"
    />
<Button android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第四个按钮"
        android:layout_column="1"
    />
</TableRow>
<TableRow>
    <Button android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第五个按钮"
        android:layout_column="2"
    />
</TableRow>
```

执行后的效果如图 1-6 所示。

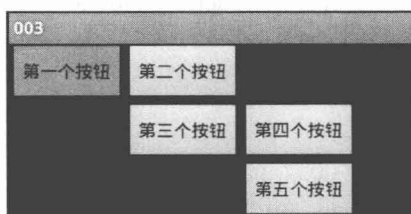


图 1-6 执行效果



实例 004：使用绝对布局（AbsoluteLayout）来布局屏幕

源码路径：daima\004

知识点介绍

绝对布局（AbsoluteLayout）是指一个 ViewGroup 以绝对方式显示它的子视图（View）元素，即以坐标的方式来定位在屏幕上位置。这种布局方式很好理解，在布局文件中编程设置 View 的

坐标，从而绝对地定位。

AbsoluteLayout 可以让子元素指定准确的 x 和 y 坐标值，并显示在屏幕上。其中坐标(0, 0)为左上角，当向下或向右移动时，坐标值将变大。AbsoluteLayout 没有页边框，允许元素之间互相重叠（尽管不推荐）。我们通常不推荐使用 AbsoluteLayout，除非有正当理由使用它，因为它会使界面代码太过刚性，以至于在不同的设备上可能不能很好地工作。AbsoluteLayout 效果如图 1-7 所示。

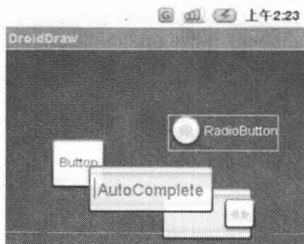


图 1-7 AbsoluteLayout 效果

具体实现

- step 1** 使用 Eclipse 创建一个名为“004”的 Android 工程。
- step 2** 编写布局文件“res/layout/main.xml”，代码如下所示。

```
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/AbsoluteLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView android:id="@+id/txtIntro"
        android:text="使用绝对布局"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_x="20dip"
        android:layout_y="20dip">
    </TextView>
</AbsoluteLayout>
```

执行后的效果如图 1-8 所示。

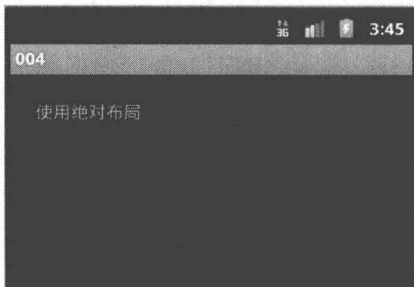


图 1-8 执行效果



实例 005：使用标签布局（TabLayout）来布局屏幕

源码路径：daima\005

知识点介绍

标签布局（TabLayout）是一个 ViewGroup 以标签的方式显示它的子视图（View）元素，就像在 Firefox 中的一个窗口中显示多个网页一样。为了创建一个标签 UI（tabbed UI），需要使用 TabHost 和 TabWidget。TabHost 必须是布局的根节点，它包含显示标签的 TabWidget 和显示标签内容的 FrameLayout。

具体实现

step 1 使用 Eclipse 创建一个名为“005”的 Android 工程。

step 2 编写布局文件“res/layout/main.xml”，代码如下所示。

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent">
<TableRow>
    <Button android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:text=" 7 " />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 8 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 9 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" / "
    />
</TableRow>
<TableRow>
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 4 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 5 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" 6 "
    />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text=" * "
    />
</TableRow>
```