Android
程序设计实用教程

左　军　编著

清华大学出版社

Android
程序设计实用教程

左　军　编著

清华大学出版社
北京

内 容 简 介

本书循序渐进地对 Android 进行介绍，内容详细、充实，实例丰富、典型。本书对每个知识点都进行了详尽的描述并为每个知识点给出对应的实例说明，让读者更容易上手，对 Android 的学习会更快捷。

本书共 9 章，主要内容包括 Android 基础知识、Android 界面设计、Android 控件设计、Android 对话框与菜单、Android 视图、Android 动画、Android 绘图、Android 数据存储与共享、Android 经典应用。通过本书的学习，读者能够在较短的时间内熟悉 Android，并掌握 Android。

本书适合没有 Android 知识基础的读者，初、中级程序员以及 Android 爱好者阅读，也可供从事 Android 开发的研究人员、工作人员、高等院校相关专业的学生使用。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989　13701121933

图书在版编目(CIP)数据

Android 程序设计实用教程/左军编著. --北京：清华大学出版社，2015
ISBN 978-7-302-39363-4

Ⅰ. ①A…　Ⅱ. ①左…　Ⅲ. ①移动终端－应用程序－程序设计－高等学校－教材　Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字(2015)第 031614 号

责任编辑：魏江江　王冰飞
封面设计：何凤霞
责任校对：时翠兰
责任印制：杨　艳

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795954

印 刷 者：北京市清华园胶印厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：28.5　　**字　　数**：694 千字
版　　次：2015 年 5 月第 1 版　　**印　　次**：2015 年 5 月第1 次印刷
印　　数：1～2000
定　　价：49.50 元

产品编号：062780-01

前　言

Android 是 Google 公司推出的专为移动设备开发的平台。从 2007 年 11 月 5 日推出以来，在短短的几年时间里就超越了称霸 10 年的诺基亚 Symbian 系统和前几年崛起的苹果 iOS 系统，成为全球最受欢迎的智能手机平台。应用 Android 不仅可以开发在手机或平板电脑等移动设备上运行的工具软件，而且可以开发 2D 甚至 3D 游戏。

从技术角度而言，Android 与 iPhone 相似，采用 WebKit 浏览器引擎，具备触摸屏、高级图形显示和上网功能，用户可以在手机上查收电子邮件、搜索网址和观看视频节目等。Android 手机比 iPhone 等其他手机更强调搜索功能，界面更丰富，可以说是一种融入了全部 Web 应用的平台。Android 的版本包括 Android 1.1、Android 1.5、Android 1.6、Android 2.0……当前的最新版本是 4.4.x。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 GoogleMap 到现在的导航系统，从基本的网页浏览到现在的 HTML5，都说明 Android 已经逐渐稳定，而且功能越来越大。此外，Google 平台不仅支持 Java、C、C++等主流编程语言，还支持 Ruby、Python 等脚本语言，甚至 Google 专为 Android 的应用开发推出了 Simple 语言，这使得 Android 有着非常广泛的开发群体。

虽然 Android 是优秀的移动操作系统，但是其程序开发的学习之旅却很艰难，最大的困难就是相关资料的缺乏。Android 是完全开源的，但不是每个程序设计人员都有时间和精力去研究它的源代码。Google 提供的主要学习资料就是 Android SDK 文档。SDK 文档对于开发人员了解 Android 程序设计有很大的帮助，但并没有系统地讲解 Android 程序设计的相关技术。针对所存在的问题，本书就此诞生。

本书通过对 Android 程序设计基础知识和基本技能进行全面系统的讲解，使读者能够轻松地掌握 Android 程序设计的基本知识和技能，尽量减少在 Android 程序设计入门阶段的摸索和徘徊，为学习 Android 程序高级技术打下基础。本书具有以下特色：

1. 结构合理

从用户的实际出发，科学安排知识内容，内容由浅入深，叙述清晰，并附加相应的实例进行操作，具有很强的知识性和实用性，反映了当前 Android 网络开发技术的发展和应用水平。

2. 通俗易懂

内容条理清晰、语言简练，可以帮助读者快速掌握每个知识点；每个部分既相互连贯又自成一体，使读者既可以按照本书编排的章节进行学习，也可以根据自己的需求对某一章节进行针对性学习。

3. 实用性强

本书彻底摒弃枯燥的理论和简单的操作，注重实用性和可操作性，将 Android 网络开发技术的理论融合到实际的操作环境中，使用户在掌握相关操作技能的同时，还能够学习到相

应的开发知识。

4. 实例丰富

书中的实例应用全面，涵盖了 Android 所能触及的领域。实例代码翔实、规范工整，且代码注释得当。

5. 图文并茂

针对没有接触过 Android 的读者，本书对相关概念一般会插入对应的图片做说明，同时几乎对每一个知识点实例都给出相应的运行效果图，这样对读者掌握这一知识点起到了很大的帮助作用。

本书共分为 9 章，其主要内容为：

第 1 章进入 Android，主要包括 Android 基本知识、Android 开发环境搭建、Android 应用组成以及 Android 模拟器操作等内容。

第 2 章 Android 界面设计，主要包括 Android 的 UI 界面、Android 布局管理器以及自定义 View 等内容。

第 3 章 Android 控件设计，主要包括 Android 文本类控件、按钮控件、编辑框控件以及条类控件等内容。

第 4 章 Android 对话框与菜单，主要包括 Android 对话框类型、Android 各类菜单等内容。

第 5 章 Android 视图，主要包括 Android 图像视图、网格视图、画廊视图、多页视图等内容。

第 6 章 Android 动画，主要包括 Android 帧动画、补间动画、动画渲染器以及动画组件等内容。

第 7 章 Android 绘图，主要包括 Android 2D 绘图及 3D 绘图等内容。

第 8 章 Android 数据存储与共享，主要包括 SharedPreferences 存储数据、File 存储数据、SQLite 存储数据以及 ContentProvider 数据共享等内容。

第 9 章 Android 经典应用，主要包括 Android 多媒体技术、无线网络、通信、定位等内容。

本书适合对象主要有以下几类：

- Android 入门级开发人员；
- 初、中级程序员；
- 培训班学生；
- Android 爱好者；
- 从事 Android 开发的研究人员和工作人员；
- 高等院校相关专业的学生。

本书主要由左军编写，此外参加编写的还有刘超、邓俊辉、梁朗星、李旭波、张棣华、刘泳、邓耀隆、梁志成和周品。

由于作者的水平有限，加之时间紧凑，书中难免存在不足之处，敬请广大读者批评指正。

编　者

2015 年 1 月

此为试读，需要完整PDF请访问：www.ertongbook.com

目 录

第 1 章 进入 Android …… 1

1.1 揭开 Android 的面纱 …… 1

1.1.1 Android 体系结构 …… 1

1.1.2 Android 自身特性 …… 3

1.1.3 Android 应用组件 …… 4

1.2 Android 开发环境 …… 6

1.2.1 Android 系统需求 …… 6

1.2.2 Android 环境搭建 …… 6

1.2.3 Eclipse 环境 …… 8

1.2.4 Android 的 ADK …… 9

1.2.5 Android 的 AVD …… 10

1.3 Android 应用组成 …… 14

1.4 第 1 个 Android 程序 …… 17

1.4.1 Android 开发流程 …… 18

1.4.2 创建应用程序 …… 18

1.5 DDMS 使用 …… 24

1.6 Android 模拟器 …… 28

1.6.1 Android 虚拟设备 …… 28

1.6.2 Android 模拟器限制 …… 29

1.6.3 Android 模拟器按键 …… 29

1.7 Android 模拟器操作 …… 30

1.7.1 删除模拟器 …… 30

1.7.2 设置语言 …… 30

1.7.3 设置输入法 …… 32

1.7.4 设置日期时间 …… 33

1.8 应用 …… 34

第 2 章 Android 界面设计 …… 36

2.1 UI 界面 …… 36

2.1.1 XML 布局控制 UI 界面 …… 36

2.1.2 Java 代码控制 UI 界面 …… 38
2.1.3 XML 和 Java 混合控制 UI 界面 …… 41
2.2 自定义 View …… 42
2.3 Android 布局管理 …… 47
2.3.1 Android 线性布局 …… 47
2.3.2 Android 表格布局 …… 50
2.3.3 Android 帧布局 …… 54
2.3.4 Android 相对布局 …… 56
2.4 Android 基本布局综合实例 …… 59
2.5 Android 其他布局 …… 65
2.5.1 Android 网格布局 …… 65
2.5.2 Android 切换卡 …… 67

第 3 章 Android 控件设计 …… 71

3.1 Widget 控件实例 …… 71
3.2 Android 文件类控件 …… 73
3.2.1 Android 文本框 …… 73
3.2.2 Android 编辑框 …… 77
3.2.3 Android 自动提示文本框 …… 80
3.2.4 Android 多选项自动提示文本框 …… 84
3.3 Android 按钮类控件 …… 86
3.3.1 Android 普通按钮 …… 86
3.3.2 Android 图片按钮 …… 89
3.3.3 Android 单选按钮 …… 91
3.3.4 Android 复选按钮 …… 94
3.3.5 CheckedTextView 控件 …… 97
3.3.6 Android 开关按钮 …… 103
3.4 Android 列表类控件 …… 105
3.4.1 Android 列表选择框 …… 105
3.4.2 Android 文本列表框 …… 109
3.5 Android 条类控件 …… 118
3.5.1 Android 进度条 …… 118
3.5.2 Android 滚动条 …… 122
3.5.3 Android 拖动条 …… 126
3.5.4 Android 星级评分条 …… 129
3.6 Android 时钟控件 …… 131
3.7 Android 日期时间控件 …… 135
3.7.1 Android 日期选择控件 …… 135
3.7.2 Android 时间选择控件 …… 137

3.8 Android 计时器 …… 139
3.9 Android 控件综合实例 …… 142

第 4 章 Android 对话框与菜单 …… 148

4.1 Android 对话框 …… 148
4.1.1 对话框概述 …… 148
4.1.2 AlertDialog 类对话框 …… 149
4.2 Android 提示框 …… 158
4.2.1 Android 消息提示框 …… 158
4.2.2 Android 状态栏上的通知 …… 163
4.3 Android 闹钟设置 …… 167
4.4 Android 菜单 …… 172
4.4.1 Android 选项菜单 …… 172
4.4.2 Android 子菜单 …… 178
4.4.3 Android 上下文菜单 …… 182
4.4.4 Android 菜单综合实例 …… 184

第 5 章 Android 视图 …… 189

5.1 Android 图像视图 …… 189
5.2 Android 网格视图 …… 193
5.3 Android 可扩展列表组件 …… 201
5.4 Android 图像切换器 …… 205
5.5 Android 画廊视图 …… 210
5.6 Android 网页浏览视图 …… 213
5.7 Android 多页视图 …… 218
5.8 Android 切换列表 …… 221
5.9 Android 滑动式抽屉 …… 224
5.10 Android 点阵图像 …… 226
5.11 Android 视图综合实例 …… 229

第 6 章 Android 动画 …… 234

6.1 Android 帧动画 …… 234
6.2 Android 补间动画 …… 239
6.2.1 Android 图像旋转 …… 240
6.2.2 Android 图像缩放 …… 243
6.2.3 Android 倾斜图像 …… 246
6.2.4 Android 图像平移 …… 249
6.2.5 Android 透明度渐变 …… 251
6.2.6 Android 补间动画的综合实例 …… 254

6.2.7 Android 自定义补间动画 …… 257
6.3 Android 帧动画与补间动画综合实例 …… 261
6.4 Android 动画渲染器 …… 264
6.5 Android 动画组件 …… 268
6.5.1 ViewSwitcher 组件 …… 268
6.5.2 ViewFlipper 组件 …… 274
6.6 SurfaceView 实现动画 …… 278
6.6.1 SurfaceView 绘制机制 …… 278
6.6.2 利用 SurfaceView 开发示波器 …… 281
6.7 Android 图像扭曲 …… 285

第 7 章 Android 绘图 …… 288

7.1 Android 常用绘图 …… 288
7.1.1 Paint 类 …… 288
7.1.2 Canvas 类 …… 291
7.1.3 Bitmap 类 …… 297
7.1.4 BitmapFactory 类 …… 298
7.2 Path 类 …… 301
7.2.1 Android 绘制文本 …… 303
7.2.2 Android 绘制图片 …… 307
7.2.3 Path 类综合实例 …… 312
7.3 Android 3D 图形 …… 318
7.3.1 OpenGL 概述 …… 318
7.3.2 Android 构建 3D 图形 …… 318
7.3.3 Android 纹理贴图 …… 324
7.3.4 Android 3D 旋转 …… 330
7.3.5 Android 3D 光照 …… 332
7.3.6 Android 3D 透明度 …… 335

第 8 章 Android 数据存储与共享 …… 338

8.1 SharedPreferences 存储数据 …… 338
8.2 File 存储数据 …… 346
8.2.1 openFileOutput、openFileInput 读/写文件 …… 346
8.2.2 SD 卡读/写文件 …… 349
8.3 SQLite 存储数据 …… 356
8.4 ContentProvider 数据共享 …… 362
8.4.1 数据模型 …… 362
8.4.2 URI 用法 …… 363
8.4.3 ContentProvider 详析 …… 368

第 9 章　Android 经典应用 …… 375

9.1　Android 多媒体技术 …… 375
9.1.1　Android 音频 …… 375
9.1.2　Android 后台播放音频 …… 382
9.1.3　Android 声音录制 …… 386
9.1.4　Android 视频 …… 390
9.1.5　Android 相机 …… 399
9.2　Android 无线网络 …… 406
9.3　Android 通信 …… 413
9.3.1　Android 语音通话 …… 413
9.3.2　Android 短信 …… 424
9.3.3　Android 电子邮件 …… 434
9.4　Android 定位 …… 436

参考文献 …… 442

第1章 进入 Android

在快速发展的移动开发领域中，以 Android 的发展最为迅猛。仅仅短短的几年，就撼动了诺基亚的霸主地位。通过在线市场，Android 的程序员不仅能向全世界贡献自己的程序，而且也能通过销售获得不菲的收入。

1.1 揭开 Android 的面纱

Android 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，例如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发。第一部 Android 智能手机发布于 2008 年 10 月。Android 逐渐扩展到平板电脑及其他领域上，例如电视、数码相机、游戏机等。2011 年的第一季度，Android 在全球的市场份额首次超过塞班系统，跃居全球第一。2013 年的第四季度，Android 平台手机的全球市场份额已经达到 78.1%。2014 年 9 月 24 日，谷歌开发的操作系统 Android 在迎来了 6 岁生日时，全世界采用这款系统的设备数量已经达到 15 亿台。

2014 年的第一季度 Android 平台已占所有移动广告流量来源的 42.8%，首度超越 iOS。但运营收入还不及 iOS。

1.1.1 Android 体系结构

Android 系统是以 Linux 系统为基础的，Google 按照功能特性将其划分为 4 层，自下而上分别是 Linux 内核、中间件、应用程序框架和应用程序，如图 1-1 所示。

1. 应用程序

Android 系统内置了一些常用的应用程序，包括 Home 视图、联系人、电话、浏览器等。这些应用程序和用户自己编写的应用程序是完全并列的，同样都是采用 Java 语言编写的。而且，用户可以根据需要增加自己的应用程序，或者替换系统自带的应用程序。

2. 应用程序框架

应用程序框架提供了程序开发人员的接口，这是与 Android 程序员直接相关的部分，开发者可以用它开发应用程序。

- 丰富而又可扩展的视图(Views)：可以用来构建应用程序，它包括列表(lists)、网格(grids)、文本框(text boxes)、按钮(buttons)，甚至可嵌入的 Web 浏览器。
- 内容提供器(Content Providers)：使得应用程序可以访问另一个应用程序的数据(如联系人数据库)，或者共享它们自己的数据。
- 资源管理器(Resource Manager)：提供非代码资源的访问，如本地字符串、图形、布

此为试读，需要完整PDF请访问：www.ertongbook.com

图 1-1　Android 系统框架图

局文件(layoutfiles)。

- 通知管理器(Notification Manager)：使得应用程序可以在状态栏中显示自定义的提示信息。
- 活动管理器(Activity Manager)：用来管理应用程序生命周期并提供常用的导航回退功能。

3. 中间件

中间件包括核心库(libraries)和 Android 运行时环境(Android runtime)两部分。

1) 核心库

核心库中主要包括一些 C/C++核心库，方便开发者进行应用的开发。

- 系统 C 库(libc)：专门为基于 embedded linux 的设备定制的。
- 媒体库：支持多种常用的音频、视频格式回放和录制，同时支持静态图像文件。编码格式包括 MPEG4、H. 264、MP3、AAC、AMR、JPG、PNG。
- SurfaceManager：对显示子系统的管理，并且为多个应用程序提供了 2D 和 3D 图层的无缝融合。
- Webkit/LibWebCore：Web 浏览引擎，支持 Android 浏览器和一个可嵌入的 Web 视图。
- SGL：底层的 2D 图形引擎。
- 3D libraries：基于 OpenGL ES 1.0 APIs 实现的 3D 引擎。
- FreeType：位图(bitmap)和矢量(vector)字体显示。
- SQLite：轻型关系型数据库引擎。

2) Android 运行时环境

Android 运行时环境主要包括以下两部分。

(1) Android 核心库：提供了 Java 库的大多数功能。

(2) Dalvik 虚拟机：依赖于 Linux 内核的一些功能，例如线程机制和底层内存管理机制。同时虚拟机是基于寄存器的，Dalvik 采用简练、高效的 byte code 格式运行，它能够在低资耗和没有应用相互干扰的情况下并行执行多个应用，每一个 Android 应用程序都在它自己的进程中运行，都拥有一个独立的 Dalvik 虚拟机实例。Dalvik 虚拟机中可执行文件为 .dex 文件，该格式文件针对小内存使用做了优化。所有的类都经由 Java 编译器编译，然后通过 SDK 中的 dx 工具转化成 .dex 格式由虚拟机执行。

4. Linux 内核

Android 平台运行在 Linux 2.6 之上，其 Linux 内核部分相当于手机硬件层和软件层之间的一个抽象层。Android 的内核提供了显示驱动、摄像头驱动、闪存驱动、键盘驱动、WiFi 驱动、音频驱动和电源管理等多项功能。此外，Android 为了让 Android 程序可以用于商业目的，将 Linux 系统中受 GNU 协议约束的部分进行了取代。

1.1.2 Android 自身特性

Android 是一种开源操作系统，其在手机操作系统领域的市场占有率已经超过了 50%，是什么原因让 Android 操作系统如此受欢迎呢？因为有其自身的几大优势。

1. 开放性

在优势方面，Android 平台首先就是开放性，开放的平台允许任何移动终端厂商加入到 Android 联盟中来。显著的开放性可以使其拥有更多的开发者，随着用户和应用的日益丰富，一个崭新的平台也将很快走向成熟。

开放性对于 Android 的发展而言，有利于积累人气，这里的人气包括消费者和厂商，而对于消费者来讲，最大的受益正是丰富的软件资源。开放的平台也会带来更大竞争，如此一来，消费者将可以用更低的价位购得心仪的手机。

2. 不受束缚

在过去很长的一段时间，特别是在欧美地区，手机应用往往受到运营商的制约，使用什么功能接入什么网络，几乎都受到运营商的控制。自从 2007 年 iPhone 上市后，用户可以更加方便地连接网络，运营商的制约减少。随着 EDGE、HSDPA 这些 2G 至 3G 移动网络的逐步过渡和提升，手机随意接入网络已不是运营商口中的笑谈。

3. 丰富的硬件

这一点还是与 Android 平台的开放性相关，由于 Android 的开放性，众多的厂商会推出千奇百怪、功能特色各具的多种产品。功能上的差异和特色，却不会影响到数据同步，甚至软件的兼容，如同从诺基亚 Symbian 风格手机一下改用苹果 iPhone，同时还可将 Symbian 中优秀的软件带到 iPhone 上使用，联系人等资料更是可以方便地转移。

4. 方便开发

Android 平台提供给第三方开发商一个十分宽泛、自由的环境，不会受到各种条条框框的阻碍，可想而知，会有多少新颖别致的软件诞生。但也有其两面性，血腥、暴力、情色方面的程序和游戏如何控制正是留给 Android 的难题之一。

5. Google 应用

在互联网的 Google 已经走过 10 年的历史，从搜索巨人到全面的互联网渗透，Google 服务如地图、邮件、搜索等已经成为连接用户和互联网的重要纽带，而 Android 平台手机将

无缝结合这些优秀的 Google 服务。

1.1.3 Android 应用组件

Android 开发四大组件分别是活动(Activity)：用于表现功能。服务(Service)：后台运行服务，不提供界面呈现。广播接收器(BroadcastReceiver)：用于接收广播。内容提供商(Content Provider)：支持在多个应用中存储和读取数据，相当于数据库。

1. 活动

在 Android 中，Activity 是所有程序的根本，所有程序的流程都运行在 Activity 之中，Activity 可以算是开发者遇到的最频繁，也是 Android 当中最基本的模块之一。在 Android 的程序当中，Activity 一般代表手机屏幕的一屏。如果把手机比作一个浏览器，那么 Activity 就相当于一个网页。在 Activity 中可以添加一些 Button、Checkbox 等控件，可以看到 Activity 概念和网页的概念类似。

一般一个 Android 应用是由多个 Activity 组成的。这多个 Activity 之间可以进行相互跳转，例如，单击一个 Button 按钮后，可能会跳转到其他的 Activity。和网页跳转稍微有些不一样的是，Activity 之间的跳转有可能返回值，例如，从 Activity A 跳转到 Activity B，那么当 Activity B 运行结束的时候，有可能会给 Activity A 一个返回值。这样做在很多时候是相当方便的。

Android 4 种的 Activity 加载模如图 1-2 所示。

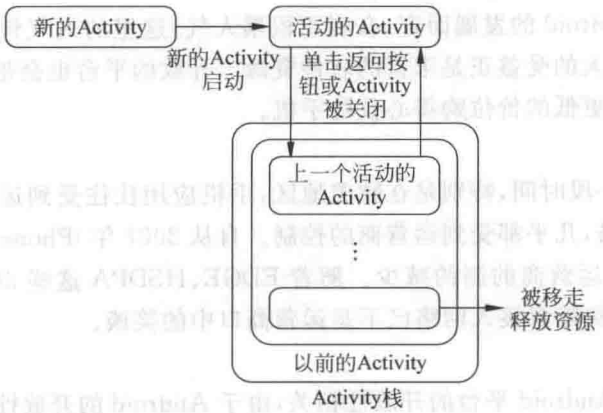

图 1-2　Android 4 种 Activity 加载模流程图

当打开一个新的屏幕时，之前的一个屏幕会被置为暂停状态，并且压入历史堆栈中。用户可以通过回退操作返回到以前打开过的屏幕。可以选择性地移除一些没有必要保留的屏幕，因为 Android 会把每个应用的开始到当前的每个屏幕保存在堆栈中。

2. 服务

Service 是 Android 系统中的一种组件，它跟 Activity 的级别差不多，但是它不能自己运行，只能在后台运行，并且可以和其他组件进行交互。Service 是没有界面的长生命周期的代码。Service 是一种程序，它可以运行很长时间，但是它却没有用户界面。这么说有点枯燥，来看一个例子。打开一个音乐播放器的程序，这个时候如果想上网了，那么，打开 Android 浏览器，这个时候虽然已经进入了浏览器这个程序，但是，歌曲播放并没有停止，而

是在后台继续一首接着一首的播放。其实这个播放就是由播放音乐的 Service 进行控制。当然这个播放音乐的 Service 也可以停止，例如，当播放列表里边的歌曲都结束了，或者用户单击了停止音乐播放的快捷键等。Service 可以在和多场合的应用中使用，例如播放多媒体的时候用户启动了其他 Activity，这个时候程序要在后台继续播放，例如检测 SD 卡上文件的变化，再或者在后台记录地理信息位置的改变，等等。

开启 Service 有以下两种方式。

(1) Context. startService()：Service 会经历 onCreate→onStart(如果 Service 还没有运行，则 Android 先调用 onCreate()，然后调用 onStart()；如果 Service 已经运行，则只调用 onStart()，所以一个 Service 的 onStart 方法可能会重复调用多次)；StopService 的时候直接 onDestroy，如果是调用者自己直接退出而没有调用 StopService 的话，Service 会一直在后台运行。该 Service 的调用者再启动后可以通过 StopService 关闭 Service。注意，多次调用 Context. startservice()不会嵌套(即使会有相应的 onStart()方法被调用)，所以无论同一个服务被启动了多少次，一旦调用 Context. stopService()或者 StopSelf()，它都会被停止。补充说明：传递给 StartService()的 Intent 对象会传递给 onStart()方法。调用顺序为：onCreate →onStart(可多次调用) →onDestroy。

(2) Context. bindService()：Service 会经历 onCreate()→onBind()，onBind 将返回给客户端一个 IBind 接口实例，IBind 允许客户端回调服务的方法，例如得到 Service 运行的状态或其他操作。这个时候调用者(Context，例如 Activity)会和 Service 绑定在一起，而 Context 退出，接着 Service 就会调用 onUnbind→onDestroyed 退出，所谓绑定在一起即为“共存亡”。

3. 广播接收器

在 Android 中，Broadcast 是一种广泛运用的在应用程序之间传输信息的机制。而 BroadcastReceiver 是对发送出来的 Broadcast 进行过滤接收并响应的一类组件。可以使用 BroadcastReceiver 来让应用对一个外部的事件做出响应。这是非常有意思的，例如，当电话呼入这个外部事件到来的时候，可以利用 BroadcastReceiver 进行处理。例如，当下载一个程序成功完成的时候，仍然可以利用 BroadcastReceiver 进行处理。BroadcastReceiver 不能生成 UI，也就是说对于用户来说不是透明的，用户是看不到的。BroadcastReceiver 通过 NotificationManager 来通知用户这些事情发生了。BroadcastReceiver 既可以在 AndroidManifest. xml 中注册，也可以在运行时的代码中使用 Context. registerReceiver()进行注册。只要是注册了，当事件来临的时候，即使程序没有启动，系统也会在需要的时候启动程序。各种应用还可以通过使用 Context. sendBroadcast()将它们自己的 Intent Broadcasts 广播给其他应用程序。

4. 内容提供商

Content Provider 是 Android 提供的第三方应用数据的访问方案。

在 Android 中，对数据的保护是很严密的，除了放在 SD 卡中的数据，一个应用所持有的数据库、文件等内容，都是不允许其他应用直接访问的。Android 当然不会真的把每个应用都做成一座“孤岛”，它为所有应用都准备了一扇窗，这就是 Content Provider。如果应用想对外提供的数据，可以通过派生 Content Provider 类，封装成一枚 Content Provider，每个 Content Provider 都用一个 uri 作为独立的标识，形如 content://com. xxxxx。所有应用看

着像 REST 的样子，但实际上，它比 REST 更为灵活。和 REST 类似，uri 也可以有两种类型，一种是带 id 的，另一种是列表的，但实现者不需要按照这个模式来做，给 id 的 uri 也可以返回列表类型的数据。

1.2 Android 开发环境

“工欲善其事，必先利其器”，在学习 Android 开发前，必须先熟悉并搭建它所需要的开发环境。

1.2.1 Android 系统需求

使用 Android SDK 进行开发时有其所必需的硬件和软件需求。对于硬件方面，要求 CPU 和内存尽量大。Android SDK 全部下载大概需要占用 4.5 GB 硬盘空间。由于开发过程中需要反复重启模拟器，而每次重启都会消耗几分钟的时间(视机器配置而定)，因此使用高配置的机器能节约不少时间。

支持 Android SDK 的操作系统及其要求如表 1-1 所示。

表 1-1 Android SDK 对操作系统的要求

操作系统	要　求
Windows	Windows XP(32 位)
	Vista(32 或 64 位)
	Windows 7(32 位或 64 位)
Mac OS	10.5.8 或更新(仅支持 x86)
Linux(在 Ubuntu 的 10.04 版测试)	需要 GNU C Library(glibc)2.7 或更新 在 Ubuntu 系统上，需要 8.04 版或更新 64 位版本必须支持 32 位应用程序

对于开发环境，除了常用的 Eclipse IDE，还可以使用 Intelli J IDEA 进行开发。对于 Eclipse 在下载 Android SDK 时就自带相兼容的版本。

1.2.2 Android 环境搭建

在 Windows 平台上，搭建 Android 开发环境，首先下载并安装与开发环境相关的软件资源，这些资源主要包括 JDK、Eclipse、Android SDK 和 Development Tools 插件(ADT 插件)。

在 Android 平台上，所有应用程序都是使用 Java 语言来编写的，所以要安装 Java 开发包 JDK(Java SE Development Kit)，JDK 是 Java 开发时所必需的软件开发包。

安装 JDK 的过程比较简单，运行该程序后，根据安装提示选择安装路径，将 JDK 安装到指定的文件夹即可，默认安装目标为 C:\Program Files\Java\jdk1.6.0_10(jdk-6u10-rc2-bin-b32-windows-i586-p-12_sep_2008)。

JDK 安装完毕后，进一步要设置 Java 的环境变量，即设置 bin 和 lib 文件夹的路径。其操作步骤如下(在计算机操作系统为 Windows 7 的环境下)：

(1) 右击“计算机”,在弹出的快捷菜单中选择“属性”选项,在弹出的“系统”对话框中,单击“高级系统设置”按钮,弹出“系统属性”对话框,如图 1-3 所示。

(2) 在“系统属性”对话框的“高级”选项卡中,单击“环境变量”按钮,弹出“环境变量”对话框,如图 1-4 所示。

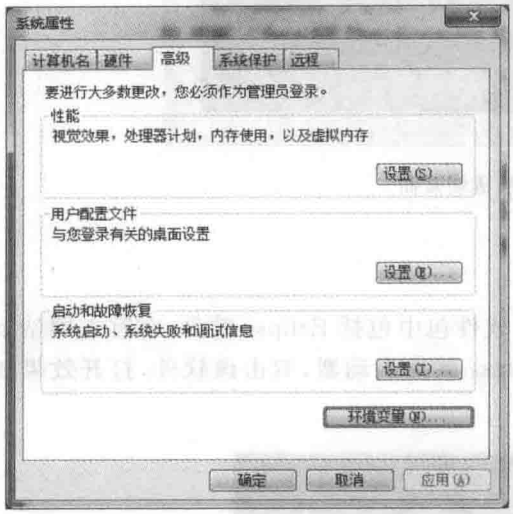

图 1-3 “系统属性”对话框

图 1-4 “环境变量”对话框

(3) 选中“系统变量”区域的 PATH 变量,单击“编辑”按钮,弹出“编辑系统变量”对话框,如图 1-5 所示。

(4) 在该对话框的“变量值”文本框中添加 C:\Program Files\Java\jdk1.6.0_10\bin,然后单击“确定”按钮即可完成设置。这样即设置了 bin 文件夹的路径。

(5) 在“环境变量”对话框的“系统变量”区域中,单击“新建”按钮,弹出“新建系统变量”对话框,如图 1-6 所示。

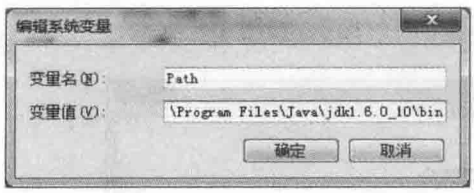

图 1-5 环境变量 Path 设置

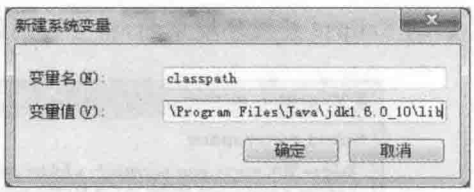

图 1-6 新建环境变量 classpath

(6) 在图 1-6 中的“变量名”右侧文本框中输入 classpath,在“变量值”右侧文本框中输入 C:\Program Files\Java\jdk1.6.0_10\lib,即可设置 lib 文件夹的路径。

完成以上操作后,一个典型的 Java 开发环境便设置好了。在正式开始下一步前先验证 Java 开发环境的设置是否成功。

在 Windows 7 系统中单击“开始”按钮,在弹出的窗口中选择“运行”,在运行框中输入 cmd 并确定,即可打开 CMD 窗口,在窗口中输入 java-version,则可显示所安装的 Java 版本信息,如图 1-7 所示。

此为试读,需要完整PDF请访问: www.ertongbook.com