

碁峯

www.gotop.com.tw

# Android



## 云计算应用开发 入门与实战

适用于  
SDK 2.X~4.X

李昇暾 詹智安 著 师蓉 改编

### 一本把 Android 技术和云计算结合的专业书

- 涵盖 Android 开发的核心技术：窗口设计组件、应用组件，Activity、Service、Broadcast Receiver 及 Content Provider 等
- 讲解了较为常用的硬件控制，如多点控制、语音、绘图、相机、GPS 定位和各种传感器，以及 Android 4.0 新增的功能和应用范例
- 用云计算案例讲解了整个开发过程，包括云平台 Hadoop 的架构、Map/Reduce 核心技术及分布式文件系统等

人民邮电出版社  
POSTS & TELECOM PRESS

# Android



## 云计算应用开发 入门与实战



李昇墩 詹智安 著 师蓉 改编

人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

Android云计算应用开发入门与实战 / 李昇瞰, 詹智安著. -- 北京: 人民邮电出版社, 2013. 7  
ISBN 978-7-115-31223-5

I. ①A… II. ①李… ②詹… 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(2013)第052469号

### 版权声明

本书为台湾基峰资讯股份有限公司独家授权的中文简化字版本。本书专有出版权属人民邮电出版社所有。在没有得到本书原版出版者和本书出版者书面许可时, 任何单位和个人不得擅自摘抄、复制本书的一部分或全部以任何形式(包括资料和出版物)进行传播。

本书原版权属基峰资讯股份有限公司。

版权所有, 侵权必究。

### 内 容 提 要

本书分5篇, 共15章, 主要介绍了用Android进行云计算开发的技术。第1篇Android基础设计篇, 讲解了Android和云计算的起源, 以及Android、Hadoop和Java的完美接合; 介绍了如何快速地打造第一个Android程序。第2篇窗口设计篇, 用两章的篇幅分别介绍了Android用户接口设计和常用窗口控件, 还特别讲解了Activity应用程序和JavaScript HTML间的互动技巧。第3篇Android移动运算的核心技术——应用组件篇, 分别深入讲解了Android的4种应用组件: Activity(活动)、Service(服务)、Broadcast Receiver(广播接收器), 以及Content Provider(内容提供者)。第4篇硬件新功能篇, 探讨了在Android开发中较为常用的硬件控制应用, 包括多点控制、语音、绘图、相机、GPS定位、各种传感器的应用等, 以及Android 4.0新增的功能和应用范例。第5篇云设计篇, 讲解了应用最广的云平台Hadoop的架构、Map/Reduce核心技术的运行原理, 以及分布式文件系统等重要议题。并用3个范例介绍了Hadoop的实战经验; 讲解了Android云决策支持系统的构建, 通过Hadoop和Android平台实现云智能的愿景。

本书适用于Android开发者、云计算开发者, 也适合作为大中专院校的教学用书和培训学校的教材。

◆ 著 李昇瞰 詹智安

改 编 师 蓉

责任编辑 张 涛

责任印制 程彦红 杨林杰

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

大厂聚鑫印刷有限责任公司印刷

◆ 开本: 800×1000 1/16

印张: 29.75

字数: 595千字

2013年7月第1版

印数: 1-3000册

2013年7月河北第1次印刷

著作权合同登记号 图字: 01-2012-4606号



定价: 79.00元

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第0021号

# 目 录

## 第 1 篇 Android 基础设计篇

第 1 章 Android 漫谈和云计算	2
1.1 Android 的起源	3
1.1.1 Android 架构	3
1.1.2 Android 历史	6
1.1.3 Android 和 Java 的甜蜜邂逅	8
1.2 云计算的起源	9
1.2.1 云计算的定义	10
1.2.2 云计算的特色	12
1.2.3 云计算的风起云涌	15
1.3 Android、Hadoop 和 Java 的完美结合	16
1.4 本章小结	16
第 2 章 我的第一个 Android 程序——HelloWorld	18
2.1 下载并安装 JDK 6	19
2.2 下载并安装 Android SDK 和 AVD Manager	20
2.3 下载并安装 Eclipse	23
2.4 安装 ADT Plugin	24
2.5 HelloWorld Android 程序设计	27
2.6 本章小结	38
第 3 章 深入探讨 HelloWorld 程序	39
3.1 Android 项目架构	40
3.2 Activity 生命周期	44
3.2.1 Android Log 机制	45
3.2.2 Activity 生命周期	48
3.3 Android 调试程序	49

## 第 2 篇 窗口设计篇

第 4 章 用户接口设计 .....	54
4.1 浅谈布局 .....	55
4.2 线性布局 .....	56
4.3 框架布局 .....	60
4.4 表格布局 .....	61
4.5 相对布局 .....	65
4.6 绝对布局 .....	66
4.7 Droid Draw 布局工具 .....	67
4.8 UI 控件的事件处理 .....	68
第 5 章 常用 UI 控件 .....	73
5.1 浅谈 UI 控件 .....	74
5.2 TextView 控件 .....	75
5.3 EditText 控件 .....	77
5.4 AutoCompleteTextView 控件 .....	80
5.5 Button 控件 .....	81
5.6 ImageView 控件 .....	82
5.7 ImageButton 控件 .....	83
5.8 RadioGroup 和 RadioButton 控件 .....	84
5.9 CheckBox 控件 .....	85
5.10 Spinner 控件 .....	87
5.11 DatePicker 和 TimePicker 控件 .....	88
5.12 AlertDialog 控件 .....	91
5.13 DatePickerDialog 和 TimePickerDialog 控件 .....	95
5.14 Toast 控件 .....	98
5.15 ProgressBar 控件 .....	98
5.16 SeekBar 控件 .....	100
5.17 RatingBar 控件 .....	101
5.18 ListActivity 和 ListView 控件 .....	102
5.19 Menu 控件 .....	104
5.20 SlidingDrawer 控件 .....	107
5.21 WebView 控件 .....	109
5.22 JavaScript 应用 .....	114

## 第 3 篇 应用组件篇

第 6 章 深入探讨 Activity 应用组件	120
6.1 单个 Activity 对应多个布局	121
6.2 多个 Activity 对应多个布局	124
6.3 再探 Activity 生命周期	128
6.4 Activity 间的值传递	132
第 7 章 数据的存储	139
7.1 SharedPreferences 存储法	140
7.2 文件存储法	145
7.3 读写外部文件法	149
7.4 SQLite 存储法	152
7.4.1 启动或创建数据库	152
7.4.2 创建数据库表	153
7.4.3 添加数据	153
7.4.4 修改数据	154
7.4.5 查询数据	154
7.4.6 删除数据	156
第 8 章 Service 应用组件	157
8.1 Service 漫谈	158
8.2 服务提供商	160
8.3 服务使用者	163
第 9 章 Broadcast Receiver 应用组件	167
9.1 Android 平台对应用程序的广播	168
9.2 应用程序间的广播	170
9.3 开启和关闭广播的接收	174
9.4 有序广播方式	177
9.5 广播通知的权限设置	181
9.6 应用程序对用户的通知	183
9.7 Broadcast 和 Notification 的整合	187
9.8 定时广播功能	189

<b>第 10 章 Content Provider 应用组件</b> .....	194
10.1 Content Provider 基本观念 .....	195
10.2 联系人数据的 Content Provider.....	197
10.2.1 添加联系人数据 .....	199
10.2.2 删除联系人数据 .....	202
10.2.3 查询联系人数据 .....	203
10.2.4 修改联系人数据 .....	205
10.3 多媒体数据的 Content Provider.....	206
10.3.1 添加图片文件 .....	206
10.3.2 删除图片文件 .....	208
10.3.3 查询图片文件 .....	208
10.3.4 修改图片文件 .....	210
10.4 自定义 Content Provider.....	212
10.4.1 添加自定义内容 .....	212
10.4.2 查询自定义内容 .....	216
10.4.3 删除自定义内容 .....	216
10.4.4 修改自定义内容 .....	217
10.5 本章小结.....	219

### 第 4 篇 硬件新功能篇

<b>第 11 章 Android 硬件控制</b> .....	222
11.1 手机相关信息 .....	223
11.2 拨号和短信发送程序.....	225
11.3 多点触控 .....	227
11.4 语音处理 .....	229
11.4.1 从文本到语音 .....	229
11.4.2 语音识别 .....	233
11.5 多媒体播放控制 .....	235
11.6 屏幕绘图 .....	244
11.6.1 View 组件绘图 .....	244
11.6.2 SurfaceView 组件绘图 .....	246
11.7 相机控制 .....	250
11.7.1 相机预览 .....	251
11.7.2 相机拍照 .....	256

11.8	定位服务	259
11.8.1	GPS 或网络定位	259
11.8.2	Google Maps 的定位服务	265
11.9	传感器使用	272
11.9.1	浅谈传感器	272
11.9.2	温度传感器	275
11.9.3	光线感应传感器	277
11.9.4	接近传感器	278
11.9.5	压力传感器	279
11.9.6	加速度传感器	280
11.9.7	重力传感器	283
11.9.8	线性加速度传感器	284
11.9.9	磁力传感器	285
11.9.10	方位传感器	285
11.10	本章小结	290
<b>第 12 章</b>	<b>Android 4.0 的新功能</b>	<b>291</b>
12.1	Android 4.0 的特色和应用程序	292
12.2	整合和新增的 API	302
12.3	Android 4.0 程序设计初探	304
12.3.1	网格布局	305
12.3.2	日历程序设计	308

## 第 5 篇 云设计篇

<b>第 13 章</b>	<b>架构 Hadoop 云系统</b>	<b>322</b>
13.1	Hadoop 漫谈	323
13.2	Hadoop 的安装和架设	325
13.2.1	安装前置环境	325
13.2.2	执行单机模式	326
13.2.3	执行伪分布式模式	329
13.3	Map/Reduce 运行原理	335
13.4	第一个 MapReduce 程序	339
13.4.1	MapReduce 程序初探	339
13.4.2	深入探讨 MapReduce 程序	342



13.5	MapReduce 相关话题	347
13.5.1	子进程 JVM 调整	347
13.5.2	运算目录结构	348
13.5.3	运算提交和监控	348
13.5.4	分布式缓存	349
13.5.5	失效管理	350
13.6	分布式文件系统	351
13.6.1	HDFS 简介	351
13.6.2	HDFS 运行架构	352
13.6.3	HDFS 副本管理	353
13.6.4	HDFS 元数据管理	355
13.6.5	HDFS 容错管理	356
13.6.6	HDFS 空间回收管理	357
13.6.7	HDFS 数据获取和程序编写	357
<b>第 14 章</b>	<b>Hadoop 分布式模式</b>	<b>363</b>
14.1	启动 Hadoop 分布式模式	364
14.2	分布式数据库系统	368
14.2.1	浅谈 HBase	369
14.2.2	数据模型	369
14.2.3	系统架构	370
14.2.4	存储架构	372
14.2.5	安装 HBase	375
14.2.6	HBase 应用程序	382
14.3	Hadoop 实战篇	392
14.3.1	最大/最小值的搜索	392
14.3.2	蒙特卡罗算法	397
14.3.3	积分求解	402
14.4	本章小结	406
<b>第 15 章</b>	<b>Android 云决策支持系统</b>	<b>407</b>
15.1	Android 网络程序设计	408
15.1.1	Android IP 程序设计	408
15.1.2	Android Web 程序设计	410
15.1.3	Android TCP/IP 程序设计	417

15.2	遗传算法	421
15.2.1	遗传算法概念	421
15.2.2	编码	423
15.2.3	种群	424
15.2.4	物竞天择	424
15.2.5	交叉	425
15.2.6	变异	427
15.2.7	演化迭代	427
15.3	云遗传算法架构	427
15.4	旅行推销员问题	430
15.5	TSP 云决策支持系统	432
15.5.1	TSP 云决策支持系统架构	432
15.5.2	TSP 云系统服务器程序	434
15.5.3	TSP 云系统客户端程序	453
15.6	本章小结	460

---

# 第 1 篇

---

## Android 基础设计篇

第 1 章 Android 漫谈和云计算

第 2 章 我的第一个 Android 程序——HelloWorld

第 3 章 深入探讨 HelloWorld 程序

---

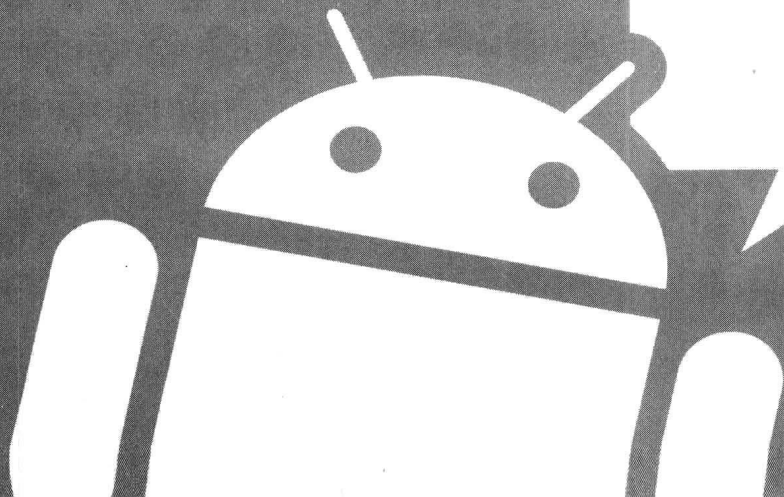
Chapter

# 1

---

## Android 漫谈 和云计算

- 1.1 Android 的起源
- 1.2 云计算的起源
- 1.3 Android、Hadoop 和 Java 的完美结合
- 1.4 本章小结



本书主要讨论 Android 和云程序设计的观念和技巧，在未进入主题之前，我们先来聊聊这两者的一些故事。

## 1.1 Android 的起源

Android（发音为：[ˈændrɔɪd]）一词最早出现于 19 世纪，法国象征主义派诗人维里耶德利尔-亚当(Villiers de L'isle Adam, 1838–1889)在 1886 年出版的《未来的夏娃(L'Eve Future)》一书中。

该书最有名的一句话就是 “If our God and our hopes are nothing but scientific phenomena, then let us admit it must be said that our love is scientific as well”，意思是说：“如果我们的神和希望都不过是科学现象而已，那么我们必须承认，我们的爱情也只是一种科学现象！”

书中的男主角为了回报他的救命恩人，帮他制造了一个女性机器人，并命名为 Hadaly，这种仿照人类制作出来的机器，在这本书中被称为 Android。今天，如果把 Android 一词当成名词使用，意思是“机器人”；如果把它当成形容词，意思是“有人类特征的”。

这种将人性、灵魂和科学之间的矛盾碰撞作为著作题材是非常吸引人的，于是当 Google 副总裁 Andy Rubin 在 2003 年成立科技公司时，便将公司命名为 Android。

究竟什么是 Android 呢？机器人？公司名称？抑或是@%\*#？简单地说，我们现在讨论的 Android 指的是一种以 Linux 为基础的、开放源代码的操作系统，刚提出时，它被设置运行在手机上。

### 1.1.1 Android 架构

Google 在 2005 年收购了 Android 公司，在 2007 年 11 月 5 日正式发布了 Android 操作系统，同年 12 月 14 日，正式版的 Android SDK 也发表问世，从此以后，工程师们便有正式的环境和工具来开发 Android 程序了。

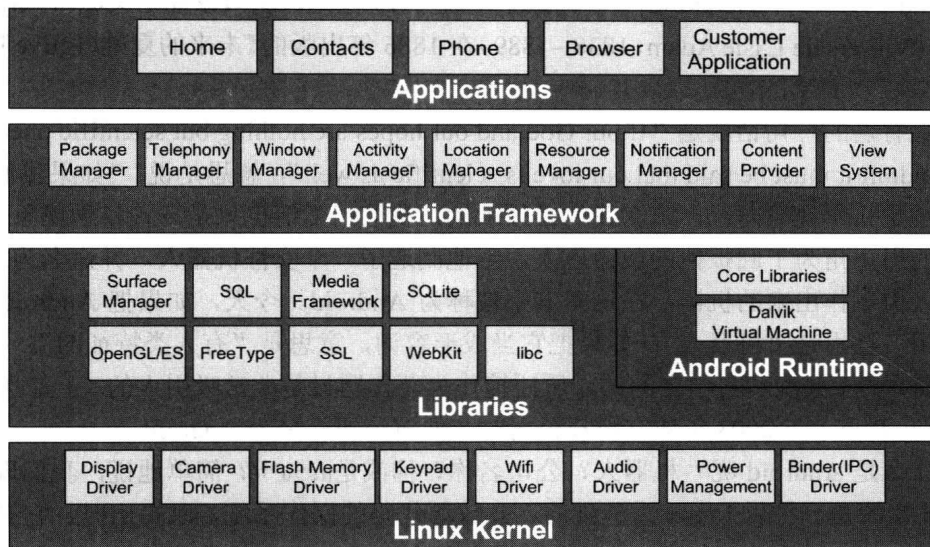
2008 年，Google 公司的 Patrick Brady 在一场介绍 Android HAL（Hardware Abstraction Layer）架构的演讲中，将题目定为 “Anatomy & Physiology of an Android”，暗喻 Android 具有生理现象，并且可供人解剖，足以展现其幽默的一面。

什么是 Android 的 HAL 架构呢？直接翻译成中文是“硬件抽象层”。简而言之，HAL 是以 so 文件的形式存在的，它可以分隔 Android 平台和 Linux kernel，让 Android 不至过度依赖 Linux kernel，以达成内核独立运行（kernel independent）的概念，同时也能在不考虑底层的驱动程序如何运行的前提下开发 Android 平台，使其自由发展。

完成这项工作并不是件容易的事，直到 2010 年 2 月 3 日，Greg Kroah-Hartman 将 Android

的驱动程序从 Linux 内核的状态树（staging tree）中移除后，Android 和 Linux 内核才真正地分道扬镳。

除了 HAL 架构外，Android 本身也具有软件堆栈（software stack）的概念，其组成包括底层的 Linux Kernel、中间层的函数库（libraries）和 Android Runtime，以及应用程序框架（application framework），最后则是最上层，由应用程序工程师开发的各种各样的应用软件，这一部分就是我们大显身手的地方。下面为 Android 软件堆栈的概念图。



在最早的时候，Linux 一词专指其内核（kernel），它提供了系统底层和硬件之间沟通的基本平台和桥梁，同时也允许其他程序可以架构在上面运行。时至今日，一般所说的 Linux 是包含 Linux kernel 以及其他软件组成的操作系统，自由软件基金会（FSF）建议将这种操作系统称为 GNU/Linux。

由于 Linux kernel 遵循 GNU General Public License version 2（GPLv2）版权，也就是所谓的 copyleft 版权。因此，除了允许使用者自由使用、散布、改作之外，Linux kernel 还要求修改后的衍生作品，也必须以同样的授权方式释出以回馈社群。

值得一提的是，选择 copyleft 授权方式，并不代表作者放弃著作权，反而是强制被授权者使用同样授权发布衍生作品，copyleft 授权条款不反对著作权的基本体制，而是通过利用著作权法进一步促进自由创作。

除此之外，GPL 的另一个特色就是算是延伸一部分的“共同运行”，指的是就算程序没有直接修改 GPL 的程序代码，但是当和某项遵循 GPL 的程序共同运行时，这样的程序也必须遵循 GPL。

在商业化情况下，如果硬件制造商希望自己的硬件能在 Linux kernel 下运行，就必须有驱动程序。遵循 GPL 就代表必须要公开驱动程序的源代码，这也等于公开硬件规格，这样所有的商业机密就会完全曝光。因此，许多硬件制造商只提供编译好的驱动程序，但不提供源代码，于是当前的硬件驱动程序都算是运行在灰色地带。

由于 Google 深知法律层面的问题，他们也不愿意系统里有什么“灰色地带”，于是采取了一些手段来避开这个问题。聪明的 Google 工程师把驱动程序移到“userspace”，简单地说，就是把驱动程序变成在 Linux kernel 上执行，而不是一起运行的东西。然后在 Linux kernel 上开个后门，让原本不能直接控制硬件的“userspace”程序也可以碰触得到，这样，只需要公布具有“后门”的 Linux kernel 的程序代码就可以了，也就可以避开 GPL 的制约了。

除此之外，Android 不支持 Cairo、X11、Alsa、Ffmpeg、GTK、Pango 及 Glibc 等；同时，用 bionic 取代 Glibc、Skia 取代 Cairo、opencore 取代 Ffmpeg 等方式和 GNU/Linux 有所区分。

再者，Google 对 Kernel 的许多修改中，最重要的就是 Binder(IPC)和 Power Management 了。Google 认为，一般的 IPC 会造成额外的资源浪费并引发安全问题，于是设计了一套专属的进程间通信（Inter-Process Communication, IPC）机制来提高其运行效率。

另外，由于便携设备的续航力向来都是最大的挑战之一，因此，Google 提供的 Power Management 在想尽办法省电。为了不影响使用时的顺畅度，Android 采取了较为积极的做法：当没有使用某项装置时，就不给该项装置提供电源，除非提出要求，才给它提供电源。因此，和一般个人计算机只有待机、休眠等状态不同，Android 细致到可以控制每一个装置的电源供应。

在软件堆栈图中的 Libraries 和 Android Runtime 共同组成了 Android 的中间层。中间层最大的作用是作为操作系统和应用程序之间的沟通桥梁，其中 Libraries 又被称为 NDK。

在 Libraries 中，Android 提供了一套特有的 SGL 函数库，它具有处理 2D 绘图的能力，相当于 Linux 的 Cairo，同时，它也是 Google Chrome 浏览器的图形引擎。此外，Android 还采用了 OpenGL ES 1.0（OpenGL for Embedded Systems）来处理 3D 绘图。最后，通过 Surface Manager 把各种要“画在”屏幕上的信息整合起来。

在 Media Framework 方面，Android 支持许多不同的多媒体格式，例如 MPEG4、H.264、MP3、AAC、AMR、JPG、GIF、PNG 等。Android 采用 OpenCORE 作为基础多媒体框架，分为 7 大块：PVPlayer、PVAuthor、Codec、PacketVideo Multimedia Framework (PVMF)、Operating System Compatibility Library (OSCL)、Common、OpenMAX。有兴趣的读者可以参考其他相关书籍来了解这部分内容的详细信息。

Libraries 中的 Webkit，其实就是 Apple Safari 浏览器背后的引擎。它可以让使用者在浏览网页时达到更好的效果，例如局部网页信息缩放、触控式操作。同时，搭配 HTML 5 可以

创造更多的可能性。在数据存储方面，Android 内建了一套 Open source 的关系数据库系统，称为 SQLite，供应用程序存储数据使用。因为是超轻量的思维设计，十分适合在便携设备上执行。

Android 中间层的另外一块是 Android Runtime，它提供了上层 Java 程序执行的运行环境。然而不同于标准的 JVM (Java Virtual Machine)，为了能更节省资源，Google 开发了独有的 Dalvik 虚拟机 (Dalvik Virtual Machine)。

Dalvik 虚拟机是一种基于寄存器 (register based) 的 Java 虚拟机，变量都存放在寄存器中，因此，虚拟机的指令相对减少，自然能够提高运行效率。除此之外，Dalvik 虚拟机也可以同时创建多个实例 (instance)，即每一个 Android 应用程序都运行在自己的 Dalvik 虚拟机上，期望能达到更好的执行效果。需要注意的是，Dalvik 虚拟机并非直接执行 Java 的字节码 (bytecode)，而是通过 DX 工具，将字节码转换成 .dex 格式，然后在 Dalvik 虚拟机上运行 .dex。

再往软件堆栈的上一层走，也就是应用程序框架层 (application framework)，这一层又被称为 SDK，它的任务是提供一系列供更上层应用软件使用的服务和系统，其中最重要的有：Activity Manager 负责管理 Activity 程序运行的生命周期、View System 提供各种各样的 GUI 控件、Content Providers 提供应用程序之间数据的共享、Resource Manager 提供非程序代码的资源存取，例如布局文件等、Notification Manager 提供在状态栏中显示和通知客户消息的功能等。

最后，也就是 Android 软件堆栈的最上层——应用软件层 (application)，它是本书要介绍的重点，同时也是应用程序工程师大显身手的地方。

对于具有 J2ME 经验的工程师来说，编写 Android 程序一点都不困难，因为其中的很多概念和 J2ME 几乎相同。例如 Activity 相当于 J2ME 的 MIDlet、View 相当于 J2ME 的 Displayable 等。(注：想进一步了解 J2ME 的读者可以参考本书姐妹篇：《Java 进阶实例设计》。)

此外，对 Android 程序的控制也和 Java Applet 基本相同，程序员必须重写很多回调函数 (call back method) 提供给底层的 Runtime 调用。而对具有 Java AWT、Swing 经验的工程师而言，事件处理 (event handling) 也完全没有入门门坎，读者可以在本书中看到很多编写监听器对象 (listener) 的实例，这就是我们熟悉的委托事件处理机制 (delegation model)。

## 1.1.2 Android 历史

Google 对 Android 具有相当大的野心，因此几乎每半年就发布新版 Android。下表整理了历年版本的简介。有趣的是，每一版本的 Android 代号都是以甜点名称来命名的，除此之外，它们还遵照 CDEFG 字母顺序排列。



Android 版本	Linux 内核版本	代号	发布日期
1.5	2.6.27	Cupcake 纸杯蛋糕	2009/04/30
1.6	2.6.29	Donut 甜甜圈	2009/09/15
2.0/2.0.1/2.1	2.6.29	Éclair 松饼	2009/10/26
2.2/2.2.1	2.6.32	Froyo 冻酸奶	2010/05/20
2.3	2.6.35	Gingerbread 姜饼	2010/12/07
3.0	2.6.36	Honeycomb 蜂巢	2011/02/02
4.0		Ice Cream Sandwich 冰淇淋三明治	2011/10/19

★ 注：Android 3.0 的代号为“蜂巢”。

为了推广 Android 平台，Google 在 2005 年以独资收购或策略联盟的方式，和多家手机制造商、软件开发商、半导体公司、电信运营商共同组成开放手机联盟（Open Handset Alliance），缩写为 OHA，期望众志成城，让 Android 有更多的用途，例如目前已经逐渐扩充功能到平板电脑领域。下表所列为 OHA 中几个较为知名的成员。

电信运营商	中国移动、中国电信、NTT DoCoMO、Sprint、T-Mobile、Vodafone 等
半导体公司	Intel、德仪、NVIDIA、ARM 等
软件开发商	Google、eBay 等
手机制造商	Acer、Asus、HTC、联想移动、NEC、Sony Ericsson、Motorola 等
商业公司	Accenture、Aplix Corporation 等

★ 注：OHA 官网，<http://www.openhandsetalliance.com>。

知名咨询公司尼尔森（Nielsen）在 2011 年 12 月初所做的调查指出，在 2011 年第 3 季，Google Android 占美国地区的智能手机市场约 42.8%，而 Apple iOS 则是 28.3%，RIM 的黑莓系统市场占有率约为 17.8%，微软的 Windows Mobile 和 Windows Phone 7 则共享 7.3% 的市场占有率。

另一项调查指出，Android 在全球 35 个国家的市场占有率已经达到 48%，欧洲达到 22.3%，日本为 57%，中国大陆为 58%，而在韩国则高达 95%（注：这或许是韩国人爱用国货的民族性使然）。