

Wrox Programmer to Programmer™



Professional Android Application Development

Android 高级编程

(英) Reto Meier 著
王鹏杰 霍建同 译



清华大学出版社

Android 高级编程

(英) Reto Meier 著

王鹏杰 霍建同 译

清华大学出版社

北 京

Reto Meier

Professional Android Application Development

EISBN: 978-0-470-34471-2

Copyright © 2009 by Wiley Publishing, Inc., Indianapolis, Indiana

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2009-4037

本书封面贴有 John Wiley & Sons 公司防伪标签, 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 高级编程/(英)梅尔(Meier, R.)著; 王鹏杰, 霍建同译. —北京: 清华大学出版社, 2010.6

书名原文: Professional Android Application Development

ISBN 978-7-302-22844-8

I. A… II. ①梅… ②王… ③霍… III. 移动通信—携带电话机—应用程序—程序设计 IV. TN929.53

中国版本图书馆 CIP 数据核字(2010)第 095094 号

责任编辑: 王 军 韩宏志

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 孟凡玉

出版发行: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

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

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

邮 购: 010-62786544

印 刷 者: 北京密云胶印厂

装 订 者: 北京鑫海金澳胶印有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 25.75 字 数: 627 千字

版 次: 2010 年 6 月第 1 版 印 次: 2010 年 6 月第 1 次印刷

印 数: 1~3000

定 价: 59.00 元

产品编号: 033388-01

作者简介

Reto Meier 出生于澳大利西南的珀斯市，现居伦敦。

Reto 是一位经验丰富的软件开发人员，拥有逾 10 年的 GUI 程序架构、设计和开发经验。他在涉足 IT 行业前曾从事过海洋石油和天然气开发以及金融工作。

Reto 始终不渝地追求掌握新技术，从 2007 年 **Android** 发布之初 **Reto** 就迷恋上了此项技术。他利用业余时间研究包括 **WPF** 在内的多种开发平台以及 **Google** 的各种开发工具。

详情请访问 **Reto** 的个人网站 **Radioactive Yak**，网址是 <http://blog.radioactiveyak.com>。

技术编辑简介

Dan Ulery 是一位在 **.NET**、**Java**、**PHP** 以及部署工程领域拥有丰富经验的软件工程师。他毕业于 **Idaho** 大学，在该校获得计算机科学理学学士学位，并辅修了数学专业。

前 言

对移动开发人员来说，现在是一个令人心潮澎湃的时代。手机从来没有像今天这样流行，强大的智能手机产品已经为消费者所普遍接受。诸如 GPS、加速计和触摸屏等外观时尚且用途广泛的手机功能硬件，正在吸引越来越多的开发者创建各种新颖有趣的移动应用程序。

虽然设计 Android 硬件的目的是为了吸引消费者，但是真正的赢家却是开发人员。由于现有的手机应用程序的开发是构建在各手机厂家的私有操作系统之上的，所以它限制了第三方应用程序的开发，而 Android 为我们提供了一个开放和通用的选择。因为有了人为制造的障碍，所以 Android 开发人员可以自由地编写能够充分利用日益强大的手机硬件的应用程序。因此，对 Android 感兴趣的开发人员都把 Google 在 2008 年发布 Android 这一举措作为移动技术发展史上的一个非常令人期待的重大事件。

由于 Android 构建在开源代码的框架之上，而且提供了强大的 SDK 库和开放的理念，所以它为广大的没有任何移动应用程序开发经验的新手开辟了一条开发完美的移动应用程序的康庄大道。而富有经验的移动开发人员现在也可以方便地扩展到 Android 平台上来，利用它独特的功能来改进现有产品或者开发其他新奇产品。

本书将指导你使用 Android 1.0 软件开发包来构建移动应用程序。在每章的讲解中，它会让你通过一系列示例项目逐步掌握 Android 中的各种新功能和技术，助你取得最圆满的学习效果。本书所介绍的各个应用实例简明扼要且极具实用价值，它们覆盖了 Android 1.0 的所有基本功能和高级功能。

由于 Android 1.0 是一个正在兴起的全新手机操作系统，所以当前支持它的手机还不多。和任何其他产品早期的发行版一样，Android 的软件和开发库还会经历很多正常的改进和完善。本书的内容和示例提供了如何使用当前 SDK 来编写优秀的移动程序所需要的基础知识，同时也保持了快速适应未来版本更强大功能的灵活性。

0.1 读者对象

本书适合所有对在 Android 移动手机平台上创建应用程序感兴趣的人。不管是对经验丰富的移动开发人员，还是想通过 Android 开发移动应用程序的新手，本书都会提供富有价值的信息。

如果读者使用过手机(特别是运行 Android 的手机)，那么这些使用经验会对阅读本书有所帮助，但这不是必需的。同样，如果以前有过手机开发经验，那么也有一定的帮助，但这也不是必需的。不过我们希望读者具有一定的软件开发经验，并且熟悉基本的开发流程。虽然了解一些 Java 知识也有用，但也不是必需的。

本书将在第 1 章和第 2 章中介绍移动应用程序开发的概要，并包含了如何在 Android

上开始开发的说明。除了这两章之外，对其他章节的阅读顺序不做要求。但是对第 3~6 章中描述的基本组件的理解，有利于你对其他章节的学习。第 7~11 章覆盖了各种可选和高级功能，你可以按照任意顺序阅读，也可以按需阅读。

0.2 本书内容

第 1 章简要介绍了 Android，包括它是什么，以及它如何适应当前的移动开发。然后详细地讲述了 Android 作为一个开放平台能够提供什么功能，并解释了它为什么是一个创建移动应用程序的良机。

第 2 章讲述了移动开发的一些最佳实践，并解释了如何下载 Android SDK 和开始开发应用程序。该章同时也介绍了 Android 开发工具，并说明了如何从头创建新的应用程序。

第 3~6 章深入地探讨了基本的 Android 应用程序组件。首先讲述了组成 Android 应用程序和它的生命周期的每个部分，在介绍“活动”内容及其生存期与生命周期之前，将简要介绍应用程序清单和外部资源。

之后将学习如何使用布局和 View 创建用户界面，并且还将了解在应用程序组件之间执行动作和发送信息的 Intent 机制。接着将介绍 Internet 资源，再后详细地讲述了数据存储、检索和共享。读者在此将了解首选保存机制、文件处理和数据库。在这个部分的最后还对如何使用内容提供者来共享应用程序数据进行了讲解。

第 7~10 章涉及到了较为高级的主题。首先介绍了地图以及基于位置的服务，接着介绍了服务、后台线程和 Notification 的用法。

接着介绍了 Android 的通信功能，包括通过即时通信和 SMS 来发送和接收消息。然后是有关硬件的内容，包括媒体录制及播放、摄像头、加速计和定位传感器。第 10 章的最后介绍了通话和网络硬件，包括通话 API、蓝牙和网络管理(包括 WiFi 和移动数据连接)。

第 11 章介绍了几个高级开发主题，其中包括安全、IPC、高级图像技术以及用户与硬件的交互。

0.3 本书结构

本书通过按照一种合理的顺序进行组织，从而帮助具有不同开发背景的读者更好地学习如何编写高级 Android 应用程序。

尽管对阅读每个章节的顺序不做要求，但是请注意，某些示例项目是跨越多个章节开发的，在其中的每个阶段都会添加一些新的功能并做一些改进。

富有移动开发经验且拥有能正常工作的 Android 开发环境的开发人员可以跳过前两章的内容——这两章简要介绍了移动开发的基本知识以及如何创建开发环境——直接学习第 3~6 章。因为这几章涵盖了 Android 开发的基础知识，所以深入地理解这几章所讲述的概

念非常重要。在学习了这几章之后，读者就可以继续学习其余的章节了，它们主要介绍了地图、基于位置的服务、后台程序以及诸如硬件交互和联网这样的更加高级的主题。

0.4 使用本书的要求

要使用本书中的示例代码，你需要通过下载 Android SDK 库和开发工具以及 Java 开发包，来创建一个 Android 开发环境。你可能还希望通过下载和安装 Eclipse 和 Android 开发人员工具插件来简化开发工作，但是这些都不是必需的。

Windows、Mac OS 和 Linux 系统环境都支持 Android 开发，可以从 Android 站点下载相应的 SDK。

要使用本书或者开发 Android 应用程序，你并不需要 Android 设备。

提示：

第 2 章更详细地列出了这些要求，并讲述了每一个组件的下载地址和安装方法。

0.5 约定说明

为了帮助你从本书中吸取到尽可能多的知识养分，并理解每部分的含义，本书将使用各种不同的约定。

提示的内容以楷体印出。

文本样式分别有：

- 使用两种不同的形式表示代码：
 - 大部分示例代码使用没有背景的等宽字体。
 - 使用灰色背景来强调当前上下文中特别重要的代码。
- 在某些示例代码中，你将看到如下的标记：

```
[... previous code goes here ...]
```

或者

```
[... implement something here ...]
```

它们表示的是使用实际的代码来代替这一整行(包括方括号)，实际代码既可能来自前面例子的代码段，也可能会在后面由读者自己实现。

0.6 源代码

读者在学习本书中的示例时，既可以手工输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/> 或

www.tupwk.com.cn/downpage 上下载。只要登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 **Download Code** 链接，就可以获得所有源代码。

提示：

由于许多图书的书名都很类似，所以按 ISBN 进行搜索是最简单的，本书英文版的 ISBN 是 978-0-470-34471-2。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

0.7 勘误表

尽管我们已经尽了最大的努力来保证文章或代码中不出现错误，但是错误总是难免的，如果你在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免走入误区，当然，这还有助于提供更高质量的信息。

要在网站上找到本书英文版的勘误表，可以登录 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 **Book Errata** 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

如果你在勘误表上没有找到错误，那么可以到 www.wrox.com/contact/techsupport.shtml 上，完成上面的表格，并把找到的错误发送给我们。我们将会核查这些信息，如果无误的话，会把它放置到本书的勘误表中，并在本书的后续版本中更正这些问题。

0.8 p2p.wrox.com

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于你张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给你传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，单击 **Register** 链接。
- (2) 阅读使用协议，并单击 **Agree** 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，并单击 **Submit** 按钮。
- (4) 你会收到一封电子邮件，其中的信息描述了如何验证账户和完成加入过程。

提示:

不加入 P2P 也可以阅读论坛上的消息, 但要张贴自己的消息, 就必须加入该论坛。

加入论坛后, 就可以张贴新消息, 回复其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要想该网站给自己发送特定论坛中的消息, 可以单击论坛列表中该论坛名旁边的 **Subscribe to this Forum** 图标。

关于使用 Wrox P2P 的更多信息, 可阅读 P2P FAQ, 了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ, 可以在任意 P2P 页面上单击 FAQ 链接。

目 录

第 1 章 Android 简介	1
1.1 一些背景信息	2
1.1.1 不远的过去	2
1.1.2 未来的前景	2
1.2 对 Android 的误解	3
1.3 开放的移动开发平台	3
1.4 自带的 Android 应用程序	4
1.5 Android SDK 功能	5
1.5.1 对包括摄像头、GPS 和加速 计在内的硬件的访问	5
1.5.2 自带的 Google 地图、地理 编码和基于位置的服务	6
1.5.3 后台服务	6
1.5.4 SQLite 数据存储和检索 数据库	6
1.5.5 共享数据和应用程序间 通信	7
1.5.6 使用 Google Talk 的 P2P 服务	7
1.5.7 扩展的数据支持和 2D/3D 图形	7
1.5.8 优化的内存和进程管理	8
1.6 开放手机联盟简介	8
1.7 运行 Android 的环境	8
1.8 从事 Android 开发的原因	9
1.8.1 推动 Android 普及的因素	9
1.8.2 Android 的独到之处	10
1.8.3 改变移动开发格局	10
1.9 开发框架简介	11
1.9.1 开发包中的资源	11
1.9.2 理解 Android 软件栈	12
1.9.3 Dalvik 虚拟机	13
1.9.4 Android 应用程序架构	14
1.9.5 Android 库	14
1.9.6 高级 Android 库	15
1.10 小结	16
第 2 章 开始入手	17
2.1 Android 开发	18
2.1.1 开始前的准备工作	18
2.1.2 创建第一个 Android 活动	22
2.1.3 Android 应用程序的类型	27
2.2 面向移动设备的开发	28
2.2.1 关于硬件设计的考虑事项	28
2.2.2 考虑用户环境	31
2.2.3 Android 开发	32
2.3 To-Do List 示例	35
2.4 Android 开发工具	40
2.4.1 Android 模拟器	41
2.4.2 Dalvik 调试监控服务 (DDMS)	41
2.4.3 Android 调试桥(ADB)	41
2.5 小结	42
第 3 章 创建应用程序和活动	43
3.1 Android 应用程序的组成部分	43
3.2 程序清单简介	44
3.3 使用清单编辑器	47
3.4 Android 应用程序生命周期	48
3.5 理解应用程序的优先级和 进程状态	49
3.6 分离资源	50
3.6.1 创建资源	50
3.6.2 使用资源	57

3.6.3	To-Do List 资源示例	60
3.6.4	为不同的语言和硬件创建资源	61
3.6.5	运行时配置更改	62
3.7	深入探讨 Android 活动	63
3.7.1	创建一个活动	63
3.7.2	活动生命周期	65
3.7.3	Android 活动类	70
3.8	小结	70
第 4 章	创建用户界面	73
4.1	Android UI 基本设计	74
4.2	View 简介	74
4.2.1	使用 View 创建 Activity (活动)用户界面	75
4.2.2	Android Widget 工具箱	76
4.3	布局简介	77
4.4	创建新的 View	78
4.4.1	修改现有的 View	79
4.4.2	创建复合控件	83
4.4.3	创建定制的 Widget 和控件	86
4.4.4	使用定制的控件	97
4.5	创建和使用菜单	98
4.5.1	Android 菜单系统简介	98
4.5.2	定义活动的菜单	100
4.5.3	动态更新菜单项	102
4.5.4	处理菜单选择	103
4.5.5	子菜单和上下文菜单	103
4.5.6	To-Do List 示例续	105
4.6	小结	111
第 5 章	Intent、广播接收器、Adapter 和 Internet	113
5.1	Intent 简介	113
5.1.1	使用 Intent 来启动活动	114
5.1.2	使用 Intent Filter 来为隐式 Intent 提供服务	121
5.1.3	使用 Intent Filter 作为插件和扩展	129
5.1.4	使用 Intent 来广播事件	132
5.2	Adapter 简介	135
5.2.1	Android 提供的部分 Adapter 简介	135
5.2.2	使用 Adapter 绑定数据	136
5.3	使用 Internet 资源	141
5.3.1	连接到 Internet 资源	142
5.3.2	利用 Internet 资源	142
5.4	Dialog 简介	143
5.4.1	Dialog 类简介	144
5.4.2	使用活动作为对话框	146
5.5	创建一个地震查看器	147
5.6	小结	156
第 6 章	数据存储、检索和共享	159
6.1	Android 中的数据保存技术	160
6.2	保存简单的应用程序数据	160
6.2.1	创建和保存 preference	160
6.2.2	检索共享的 preference	161
6.2.3	保存活动状态	162
6.2.4	为地震查看器创建一个 Preference 页	165
6.3	保存和载入文件	174
6.3.1	将静态文件作为资源添加	174
6.3.2	文件管理工具	175
6.4	Android 中的数据库	175
6.4.1	SQLite 简介	175
6.4.2	Cursor 和内容值	176
6.4.3	使用 Android 数据库	177
6.5	内容提供者简介	189
6.5.1	使用内容提供者	190
6.5.2	本地 Android 内容提供者	192
6.5.3	创建一个新的内容提供者	194
6.5.4	创建和使用地震内容提供者	197
6.6	小结	206

第 7 章 地图、地理编码和基于位置的服务 207	7.8 对 Earthquake 示例添加地图功能..... 241
7.1 使用基于位置的服务..... 207	7.9 小结..... 247
7.2 使用 Test Provider 构建模拟器..... 208	第 8 章 后台工作 249
7.2.1 更新模拟位置提供者中的位置..... 208	8.1 服务简介..... 250
7.2.2 创建一个应用程序来管理 Test Location Provider..... 209	8.1.1 创建和控制服务..... 250
7.3 选择一个 Location Provider..... 211	8.1.2 将活动和服务绑定..... 259
7.3.1 查找可用的提供者..... 212	8.2 使用后台工作线程..... 260
7.3.2 根据要求标准查找提供者..... 212	8.2.1 创建新的线程..... 260
7.4 确定自己所在的位置..... 213	8.2.2 为 GUI 操作同步线程..... 261
7.4.1 “Where Am I” 示例..... 213	8.2.3 将 Earthquake Service 移动到后台线程..... 262
7.4.2 追踪移动..... 216	8.3 创建一个 Toast..... 263
7.4.3 更新 “Where Am I” 示例中的位置..... 217	8.3.1 定制 Toast..... 263
7.5 使用邻近提醒..... 219	8.3.2 在工作(worker)线程中使用 Toast..... 265
7.6 使用 Geocoder..... 220	8.4 Notification 简介..... 266
7.6.1 反向地理编码..... 221	8.4.1 Notification Manager 简介..... 267
7.6.2 前向地理编码..... 221	8.4.2 创建 Notification..... 267
7.6.3 对 “Where Am I” 示例进行地理编码..... 222	8.4.3 触发 Notification..... 268
7.7 创建基于地图的活动..... 223	8.4.4 向 Earthquake Monitor 中添加 Notification..... 268
7.7.1 MapView 和 MapActivity 简介..... 224	8.4.5 高级 Notification 技术..... 271
7.7.2 创建一个基于地图的活动..... 224	8.4.6 持续的和连续的 Notification..... 273
7.7.3 配置和使用 Map View..... 226	8.5 使用 Alarm..... 274
7.7.4 使用 Map Controller..... 227	8.6 使用 Alarm 更新 Earthquake..... 276
7.7.5 对 “Where Am I” 示例使用地图..... 227	8.7 小结..... 278
7.7.6 创建和使用覆盖(Overlay)..... 231	第 9 章 P2P 通信 279
7.7.7 MyLocationOverlay 简介..... 238	9.1 Android 即时消息简介..... 279
7.7.8 ItemizedOverlay 和 OverlayItem 简介..... 239	9.1.1 使用 GTalk 服务..... 280
7.7.9 将 View 固定到地图和地图的某个位置上..... 240	9.1.2 和 GTalk 服务绑定..... 281
	9.1.3 建立 GTalk 连接, 开始一个 IM 会话..... 282
	9.1.4 在线状态和联系人列表简介..... 282
	9.1.5 管理聊天会话..... 285

9.1.6	发送和接收数据信息	289
9.2	SMS 简介	291
9.2.1	在应用程序中使用 SMS	291
9.2.2	发送 SMS 信息	291
9.2.3	监听 SMS 消息	294
9.2.4	紧急响应的 SMS 示例	296
9.2.5	紧急响应自动化	306
9.3	小结	314
第 10 章	访问 Android 硬件	317
10.1	使用媒体 API	317
10.1.1	播放媒体资源	318
10.1.2	录制多媒体	319
10.2	使用摄像头	321
10.2.1	控制摄像头设置	321
10.2.2	使用摄像头预览	321
10.2.3	照相	322
10.3	Sensor Manager 简介	323
10.4	使用加速计和指南针	324
10.4.1	加速计介绍	325
10.4.2	检测加速度的改变	325
10.4.3	创建一个速度计	327
10.4.4	确定方向	330
10.4.5	创建指南针和地平仪	332
10.5	Android 电话功能	335
10.5.1	打电话	335
10.5.2	监控电话状态和电话 活动	336
10.5.3	监控数据连通性和 活动	338
10.5.4	访问手机的属性和状态	339
10.5.5	控制电话	340
10.6	使用蓝牙	341
10.6.1	蓝牙服务介绍	341
10.6.2	控制本地蓝牙设备	341
10.6.3	发现并配对蓝牙设备	342
10.6.4	管理蓝牙连接	343
10.6.5	使用蓝牙进行通信	343
10.6.6	使用蓝牙耳机	345
10.7	管理网络和 Wi-Fi 连接	346
10.7.1	监控和管理 Internet 连接	346
10.7.2	管理活动的连接	347
10.7.3	管理你的 Wi-Fi	348
10.8	控制设备震动	351
10.9	小结	352
第 11 章	Android 高级开发	353
11.1	Android 的安全性	353
11.1.1	Linux 内核安全	354
11.1.2	权限简介	354
11.1.3	声明和实施权限	354
11.1.4	为广播 Intent 实施 权限	355
11.2	使用 AIDL 来支持服务 IPC	355
11.3	使用 Internet 服务	360
11.4	构建内容丰富的用户界面	361
11.4.1	使用动画	361
11.4.2	使用主题来为应用程序 添加皮肤	372
11.4.3	高级画布绘图	373
11.4.4	SurfaceView 简介	389
11.4.5	创建交互式控件	392
11.5	小结	397

第 1 章

Android 简介

无论你是一名经验丰富的移动开发工程师、一个桌面或者 Web 开发人员还是一个纯粹的编程新手，Android 都为编写具有创新性的移动应用程序带来了令人兴奋的新机遇。

虽然它被命名为 Android(机器人)，但是它并不会帮助你打造一支势不可挡的铁血机器人部队来修复地球上人类造成的破坏。事实上，Android 是一个开源的软件栈，它包含了操作系统、中间件和关键的应用程序，以及一组用于编写移动应用程序的 API 库；应用程序决定着移动设备样式、观感和功能。

小巧玲珑，外观时尚且用途广泛的现代手机已经成为了集照相机、媒体播放器、GPS 系统和触摸屏为一体的强大工具。随着技术的发展，移动设备的功能已不再仅仅是打电话那么简单，它们的软件 and 开发平台也在努力追赶着日新月异的科技时代的步伐。

不久前，手机平台在很大程度上还是一个封闭的环境，主要还是以要求使用专有开发工具的专用操作系统为基础构建的。通常这些手机自身的应用程序的优先级要比第三方开发的应用程序的优先级别高。对于那些希望在硬件性能日益强大的移动设备上进行开发的人员来说，这就设置了人为障碍。

在 Android 中，本地应用程序和第三方应用程序使用相同的 API 编写，并且在相同的运行时(run time)上执行。这些 API 的功能包括硬件访问、基于位置的服务(location-based service)、后台服务支持、基于地图的活动(activities)、关系数据库、设备间点对点通信以及 2D 和 3D 图形。

通过本书，你可以学习到怎样使用这些 API 来开发自己的 Android 应用程序。在本章中，你将会学习一些移动开发的准则以及一些在 Android 开发平台中可用的功能。

Android 拥有功能强大的 API、出色的文档、茁壮成长的开发人员社区，而且不需要为开发或发布支付费用。随着移动设备的日益普及，不管在什么样的开发背景下，使用 Android 来开发新颖的手机应用程序都是一个令人为之振奋的良机。

1.1 一些背景信息

在 Twitter 和 Facebook 出现之前，当创建 Google 的想法还被认为是一个天方夜谭的时候，刚刚问世的手机仅仅是如下这样一个概念——它们是足够小的便携电话，能够放在一个公文包中，电池足够用上几个小时，并且可以不用建立物理通信线路就能自由通信。

现在，小巧、时尚而且功能强大的手机已经相当普及并且不可或缺。硬件上的进步使手机在拥有越来越多的外围设备的同时也变得更加小巧和高效。

随着摄像头和媒体播放器的问世，现在的手机更是包含了 GPS 系统、加速计和触摸屏。虽然这些硬件上的创新为软件开发提供了广泛的应用基础，但实际情况却不容乐观，手机上实用的应用程序已经落后于相应的硬件水平了。

1.1.1 不远的过去

过去，那些通常使用 C 或者 C++ 进行编程的开发人员，必须理解在其上编写代码的特定硬件，通常是一个设备，但也可能是来自于同一家生产商的一系列设备。随着硬件技术的发展，这种封闭的方法很难追赶硬件发展的步伐。

更近一点，人们开发出了像 Symbian 这样的平台，从而给开发人员提供了更广泛的目标用户群(target audience)。这些系统在鼓励移动开发人员开发更加丰富的应用程序以便更高效地利用硬件方面，比上述那种封闭的方法更加成功。

这些平台提供了一些访问设备硬件的接口，但是要求编写复杂的 C/C++ 代码，而且严重依赖那些以难以使用而著称的私有 API。当开发那些必须运行在不同的硬件实现上的应用程序的时候，这些困难就凸现在了开发人员面前，特别是在实现某种硬件功能(如 GPS)的时候尤其如此。

近几年，移动开发的最大亮点在于引入了由 Java 承载(java-hosted)的 MIDlet。MIDlet 是在一个 Java 虚拟机上执行的，它把底层的硬件抽象出来，从而使开发人员可以开发出能运行在更多种类的硬件上的应用程序，只要这些硬件支持 Java 运行时 (Java run time) 就可以。遗憾的是，这种便利是以对设备硬件的限制为代价的。

在移动开发中，通常第三方应用程序的硬件访问和执行权限与手机制造商编写的本地 (native) 应用程序的权限是不同的。在 MIDlet 中，第三方应用程序拥有的这两种权限一般都要比本地应用程序低。

Java MIDlet 的引入扩大了开发人员的目标用户群，但是由于缺乏对低级硬件的访问权限以及沙盒式的执行等原因，这就意味着大部分移动应用程序都是运行在较小屏幕上的桌面程序，而没有充分利用手机平台的固有移动性。

1.1.2 未来的前景

最近出现的一些手机操作系统设计理念的创新与突破，都是为了适应手机硬件日益强大的计算能力，Android 也是在这样一种背景下应运而生的。现在，Windows Mobile 和 Apple 的 iPhone 为手机应用程序开发提供了一个功能更丰富、更加简捷的开发环境。然而，与

Android 不同的是，它们是构建在私有操作系统的基础上的，而私有操作系统中的本地应用程序通常比第三方创建的应用程序拥有更高的优先级，并且私有操作系统还限制了应用程序和本地手机数据之间的通信。Android 通过提供一个以开源的 Linux 内核为基础而构建的开放的开发环境，从而为移动应用程序的开发提供了新的机遇。通过一系列 API 库，所有应用程序都可以对硬件进行访问，并且可以在严格受控的条件下完全支持应用程序之间的交互。

在 Android 中，所有的应用程序地位都相等。第三方和本地的应用程序都使用相同的 API 进行编写，而且都在相同的运行时下执行。用户可以删除任何的本地应用程序，并且可以用任意一个第三方开发人员的应用程序代替，甚至连拨号程序和主屏幕都可以进行替换。

1.2 对 Android 的误解

作为对一个成熟领域破坏性的补充，我们就不难理解为什么一些人会对 Android 具体是什么这个问题存在很多疑惑。Android 不是：

- 一个 Java ME 实现 Android 应用程序是使用 Java 语言编写的，但是它们并不是运行在一个 Java ME 虚拟机上的，而且已编译的 Java 类和可执行程序不能在不经修改的情况下就运行在 Android 上。
- Linux 手机标准论坛(Linux Phone Standards Forum, LiPS)或者开放手机联盟(Open Mobile Alliance, OMA)的一部分 Android 是运行在一个开源的 Linux 内核的基础上的，尽管它们的目标很相似，但是 Android 的完全软件栈方法和这些标准定义组织所关注的重心是不同的。
- 一个简单的应用层(如 UIQ 或者 S60) 尽管 Android 并没有包含一个应用层，但是它也描述了整个软件栈，这个软件栈包含了底层操作系统、API 库和应用程序本身。
- 一个移动手机设备 Android 包含了一个移动设备制造商的参考设计，但是它和 iPhone 不同，例如，并不存在单个“Android Phone”。相反，Android 是为了支持更多可选的硬件设备而设计的。
- Google 对 iPhone 的回应 iPhone 是由 Apple 公司发布的完全私有的硬件和软件平台，而 Android 是由 OHA(Open Handset Alliance)生产和支持的一个开源的软件栈，是为了能在任何满足要求的手机上运行而设计的。现在有很多人在讨论 Google 商标的 Android 手机，但是即使 Google 生产了它，它也只是 Android 平台下一家公司的硬件实现而已。

1.3 开放的移动开发平台

Google 把 Android 描述为：

提示:

为移动设备设计的第一个真正开放的综合平台, 所有的软件都能运行在手机上, 从而消除了阻碍移动创新的屏障。

<http://googleblog.blogspot.com/2007/11/wheres-my-gphone.html>

Android 由几个不可或缺且相互独立的部分组成:

- 一个硬件参考设计, 它描述了一个移动设备为了支持软件栈而需要的性能。
- 一个 Linux 操作系统内核, 它提供了与硬件之间的低级接口、内存管理和进程控制, 并且全部都为移动设备进行了优化。
- 应用程序开发的开源库, 包括 SQLite、WebKit、OpenGL 以及一个媒体管理器。
- 用来运行和承载(host)Android 应用程序的运行时, 包括 Dalvik 虚拟机和提供 Android 特定功能的核心库。为了在移动设备上使用, 将其设计成为了小巧而高效的运行时。
- 一个把系统服务隐式地显示给应用层的应用程序框架, 包括窗口管理器、内容提供者、位置管理器、电话和点对点服务。
- 一个用来承载(host)和启动应用程序的用户接口框架。
- 作为软件栈的一部分发行的预装应用程序。
- 用来开发应用程序的软件开发包, 包括工具、插件和文档。

提示:

虽然我们期待在手机上市的时候所有的 Android 栈都是开源的, 但是在目前这个阶段, 并不是所有的 Android 栈都是开源的。当然, 你为 Android 开发的应用程序也不必是开源的。

真正使 Android 引人注目的是它的开放哲学, 这就保证了用户界面或者本地应用程序的所有不足都可以通过编写一个扩展或者替代品来解决。作为开发人员, Android 提供了一个完全按照对样式、观感和功能的想象来设计手机界面和应用程序的机会。

1.4 自带的 Android 应用程序

Android 手机在正常情况下都会有一套预装的应用程序, 包括但不限于以下几种:

- 一个兼容 Gmail、但不限于 Gmail 的邮件客户端。
- 一个 SMS 管理应用程序。
- 一个完整的 PIM(personal information management, 个人信息管理)套装, 包括日历和联系人列表, 它们都与 Google 的在线服务紧密地联系在一起。
- 一个有特色的移动 Google 地图应用程序, 包括街道浏览(StreetView)、商务搜索器(business finder)、驾驶导航(driving direction)、卫星观察(satellite view)和交通路况(traffic conditions)。
- 一个基于 WebKit 的 Web 浏览器。
- 一个即时消息客户端。