

普通高等学校规划教材

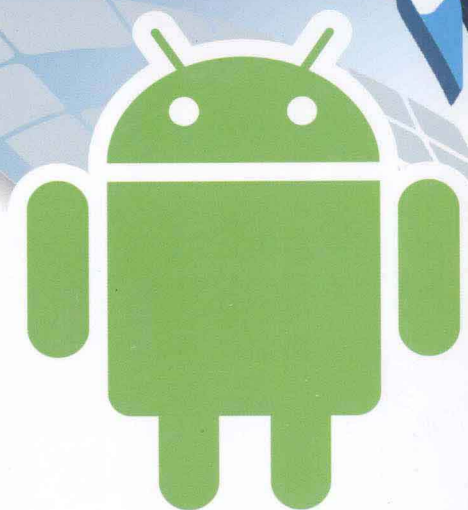
Android

编程基础

主编◎于 彬



西南交通大学出版社
[Http://press.swjtu.edu.cn](http://press.swjtu.edu.cn)



普通高等学校规划教材

Android 编程基础

主编 于 彬

西南交通大学出版社

· 成 都 ·

内容提要

本书以开发设计良好的 Android 应用为主题,详细介绍了 Android 应用程序开发所需的基础知识和基本思想,内容翔实,通俗易懂。其主要内容包括 Android 的开发环境、系统框架、用户界面开发、生命周期、组件通信与广播消息、后台服务、数据存储、定位服务与地图应用、网络编程等方面,较全面地覆盖了 Android 程序开发所涉及的内容。

本书可作为高等院校理工科相关专业本科生教材,也可作为开发人员或软件实践者自学和提高了的参考书。

图书在版编目(CIP)数据

Android 编程基础 / 于彬主编. — 成都:西南交通大学出版社, 2013.9
普通高等学校规划教材
ISBN 978-7-5643-2600-5

I. ①A… II. ①于… 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(2013)第 196350 号

普通高等学校规划教材

Android 编程基础

主编 于彬

*

责任编辑 陈斌

助理编辑 黄庆斌

封面设计 墨创文化

西南交通大学出版社出版发行

四川省成都市金牛区交大路 146 号 邮政编码: 610031 发行部电话: 028-87600564

<http://press.swjtu.edu.cn>

四川森林印务有限责任公司印刷

*

成品尺寸: 185 mm × 260 mm 印张: 18.5

字数: 462 千字

2013 年 9 月第 1 版 2013 年 9 月第 1 次印刷

ISBN 978-7-5643-2600-5

定价: 29.80 元

图书如有印装质量问题 本社负责退换
版权所有 盗版必究 举报电话: 028-87600562

前 言

Android 是 Google 发布的基于 Linux 平台的开源移动操作系统，目前已经占据了智能手机操作系统市场份额第一位。基于 Android 的移动应用开发已经成为软件开发中新的热点和发展趋势。随着 Android 的飞速发展，越来越多的厂商已参与 Android 开发与推广，越来越多的用户也选择使用 Android。与此同时，越来越多的开发者正在投入到 Android 开发阵营中。本书可以帮助对 Android 开发有兴趣的读者快速进入 Android 移动开发领域。

本书的读者应具有一定的 Java 语言基础。读者通过学习本书，能够掌握 Android 编程的基本知识和主要技术。本书章节的例子都是从简单到复杂，逐步深入，叙述上追求简单直观，通俗易懂，便于读者学习掌握 Android 编程技术。

本书是根据软件工程专业培养目标而编写的高等学校教材，突出应用型本科人才培养模式：内容体系打破科学性，突出使用性和技能性；理论部分坚持必须够用为度，以应用为主旨，强调实践、实训教学环节，以培养学生的综合实践能力。

全书以由浅入深、循序渐进的方式介绍了 Android 应用开发的相关理论知识，叙述上追求结构层次清晰、语言简洁、案例简单、易于理解，需要注意的是，本书读者应具备一定的 Java 语言基础。全书共分为 7 章：第 1 章介绍了 Android 的发展历程、Android 操作系统的整体架构，详细讲解了 Android 环境的搭建以及 Android 程序开发的步骤；第 2 章介绍了 Android 架构中比较常见的、重要的界面元素；第 3 章主要介绍了 Android 平面图形处理、动画处理、OpenGL 3D 编程技术；第 4 章介绍了 Android 应用程序的主要组件，详细介绍了 Activity 组件的生命周期，Intent 意图和服务组件；第 5 章介绍了 Android 的多种数据存储方式；第 6 章介绍了 Android 的网络编程技术；第 7 章介绍了 GPS 位置和 Google Map 在 Android 中的应用编程。

全书由于彬担任主编，具体编写分工为：第 1、6、7 章由于彬编写；第 2 章由张磊编写；第 3 章由鲁宛生编写；第 4、5 章由陈连山编写。

由于时间仓促，加之编者水平有限，书中难免存在不足之处，恳请读者批评指正。本书在编写过程中参考了 Android 官网、相关网络文章和 Android 开发书籍，在此向相关作者表示真挚的感谢。

编 者

2013 年 5 月

目 录

第 1 章	Android 开发基础	1
1.1	Android 概述	1
1.2	Android 操作系统	5
1.3	Android 开发环境	9
1.4	第一个 Android 应用程序	16
第 2 章	Android UI 基础	31
2.1	Android 平台 UI 架构	31
2.2	UI 组件的定义和引用	32
2.3	UI 基础控件	34
2.4	UI 高级控件	46
2.5	UI 布局	61
2.6	Android 事件监听器和事件处理机制	70
	本章小结	76
第 3 章	Android 图形处理	77
3.1	使用简单图片	77
3.2	Android 绘图	83
3.3	图形特效处理	91
3.4	动画	100
3.5	OpenGL 视图绘制	110
	本章小结	131
第 4 章	Android 应用程序的组件	132
4.1	Android 应用程序的组件简介	132
4.2	Activity	133
4.3	Intent 意图	146
4.4	Service 服务	153
4.5	Broadcast Receiver 广播	161
	本章小结	168
第 5 章	Android 数据存储	169
5.1	文件存储	169
5.2	SharedPreferences—轻量级数据存储	181

5.3	SQLite 数据库存储	186
5.4	ContentProvider—数据共享	201
5.5	Android 数据库打包	214
	本章小结	217
第 6 章	Android 网络编程	218
6.1	Android 网络编程简介	218
6.2	URL 网络编程	220
6.3	HTTP 网络编程	222
6.4	Socket 编程	241
6.5	WebKit For Android	251
	本章小结	265
第 7 章	GPS 位置和 Google Map	266
7.1	GPS 位置	266
7.2	Google Map	275
	本章小结	289
	参考文献	290

第 1 章 Android 开发基础

【学习目标】

本章需要了解 Android 有关技术，初步了解 Android 系统；掌握 Android 应用开发环境的配置；能够使用 Eclipse 编写一个 Android 的 Hello World 应用程序。

1.1 Android 概述

Android OS (Operating System) 是由 Google 领衔的手机联盟开发的一个移动设备、智能手机和平板电脑的操作系统。

Android 是基于 Linux 的操作系统，主要用于触摸屏的移动设备，如智能手机、平板电脑。2007 年成立的 Android 开放手机联盟，致力于推动开放的移动设备标准。2008 年 10 月，第一款 Android 系统的手机诞生并发售。

1.1.1 Android 发展概述

2007 年 11 月 5 日，Google 与其他 33 家手机制造商（摩托罗拉、宏达电、三星、LG 等）、手机晶片供应商、软硬件供应商、电信业者联合组成的开放手持装置联盟 (Open Handset Alliance)，发布了名为“Android”的开放手机软硬件平台。参与开放手持装置联盟的这些厂商，都会基于 Android 平台来开发新的手机业务。

2008 年 9 月 24 日，T-Mobile 首度公布第一台 Android 手机 (G1) 的细节，同日 Google 也推出了 Android SDK 1.0 版本。对应用程序开发者而言，1.0 代表了开发者可以放心地使用 API，而不必再担心 API 有太大的变动。G1 在同年 10 月 20 日正式发售。在发售前仅针对原 T-Mobile 用户的预购活动中，就已经被预购了 150 万台。次日，Open Handset Alliance 公开了全部 Android 的源代码。从此，开发者拥有了一个完全开放的手机平台。

1. Android 发展历程

2003 年 10 月，Andy Rubin 等人创建了 Android 公司，并组建 Android 团队。2005 年 8 月 17 日，Google 低调收购了成立仅 22 个月的高科技企业 Android 及其团队。Andy Rubin 成为 Google 公司工程部副总裁，继续负责 Android 项目。

2007 年 11 月 5 日，Google 公司正式向外界展示了这款名为 Android 的操作系统，并且在这天宣布建立一个全球性的联盟组织，该组织由 34 家手机制造商、软件开发商、电信运营商以及芯片制造商共同组成，并与 84 家硬件制造商、软件开发商及电信营运商组成开放手持设备联盟 (Open Handset Alliance) 来共同研发和改良 Android 系统，这一联盟将支持

Google 发布的手机操作系统以及应用软件，Google 以 Apache 免费开源许可证的授权方式，发布了 Android 的源代码。2008 年，在 Google I/O 大会上，Google 提出了 Android HAL 架构图。同年 8 月 18 日，Android 获得了美国联邦通信委员会（FCC）的批准。在 2008 年 9 月，Google 正式发布了 Android 1.0 版本的操作系统，这也是 Android 系统最早的版本。

2009 年 4 月，Google 正式推出了 Android 1.5 版本的操作系统。从 Android 1.5 版本开始，Google 开始将 Android 的版本以甜品的名字命名，Android 1.5 命名为 Cupcake（纸杯蛋糕）。该系统与 Android 1.0 相比有了很大的改进。

2009 年 9 月，Google 发布了 Android 1.6 的正式版，并且推出了搭载 Android 1.6 正式版的手机 HTC Hero（G3）。该手机凭借出色的外观设计以及全新的 Android 1.6 操作系统，成为当时全球最受欢迎的手機。Android 1.6 也有一个有趣的甜品名称，称为 Donut（甜甜圈）。

2. Android 各个版本在市场受欢迎度

2010 年 2 月，Linux 内核开发者 Greg Kroah-Hartman 将 Android 的驱动程序从 Linux 内核“状态树”（“staging tree”）上除去，从此，Android 与 Linux 开发主流将分道扬镳。在同年 5 月，Google 正式发布了 Android 2.2 操作系统。Google 将 Android 2.2 操作系统命名为 Froyo（冻酸奶）。

2010 年 10 月，Google 宣布 Android 系统达到了第一个里程碑，即电子市场上获得官方数字认证的 Android 应用数量已经达到了 10 万个，Android 系统的应用增长非常迅速。在 2010 年 12 月，Google 正式发布了 Android 2.3 操作系统 Gingerbread（姜饼）。

2011 年 1 月，Google 称每日的 Android 设备新用户数量达到了 30 万。到 2011 年 7 月，这个数字增长到 55 万，而 Android 系统设备的用户总数达到了 1.35 亿，Android 系统已经成为智能手机领域占有量最高的系统。

2011 年 8 月 2 日，Android 手机已占据全球智能机市场 48% 的份额，并在亚太地区市场占据统治地位，终结了 Symbian（塞班系统）的霸主地位，跃居全球第一。

2011 年 9 月，Android 系统的应用数目已经达到了 48 万，而在智能手机市场，Android 系统的占有率已经达到了 43%，继续排在移动操作系统首位。Google 将会发布全新的 Android 4.0 操作系统，这款系统被 Google 命名为 Ice Cream Sandwich（冰激凌三明治）。

2012 年 1 月 6 日，Google Android Market 已有 10 万开发者推出超过 40 万活跃的应用，大多数的应用程序为免费。Android Market 应用程序商店目录在新年首周末突破 40 万应用基准，距离突破 30 万应用仅 4 个月。在 2011 年早些时候，Android Market 从 20 万应用增加到 30 万应用也只花了 4 个月。

3. 发行版本

Android 在正式发行之之前，最开始拥有两个内部测试版本，并且以著名的机器人名称来对其进行命名，它们分别是：阿童木（Android Beta）和发条机器人（Android 1.0）。后来由于涉及版权问題，Google 将其命名规则变更为用甜点作为它们系统版本的代号的命名方法，如图 1.1 所示。甜



图 1.1 甜点命名方法

点命名法开始于 Android 1.5 发布的时候。作为每个版本代表的甜点的尺寸越变越大，然后按照 26 个字母顺序，如上面提到的纸杯蛋糕（Android 1.5），甜甜圈（Android 1.6），松饼（Android 2.0/2.1），冻酸奶（Android 2.2），姜饼（Android 2.3），蜂巢（Android 3.0），冰激凌三明治（Android 4.0），果冻豆（Jelly Bean，Android 4.1 和 Android 4.2）。

(1) Android 1.1: 2008 年 9 月发布的 Android 第 1 版。

(2) Android 1.5 Cupcake（纸杯蛋糕）: 2009 年 4 月 30 日发布。

主要更新如下:

拍摄/播放影片，并支持上传到 Youtube；支持立体声蓝牙耳机，同时改善自动配对性能；最新的采用 WebKit 技术的浏览器，支持复制/粘贴和页面中搜索；GPS 性能大大提高；提供屏幕虚拟键盘；主屏幕增加音乐播放器和相框 Widgets；应用程序自动随着手机旋转；短信、Gmail、日历，浏览器的用户接口大幅改进，如 Gmail 可以批量删除邮件；相机启动速度加快，拍摄图片可以直接上传到 Picasa；来电照片显示。

(3) Android 1.6 Donut（甜甜圈）: 2009 年 9 月 15 日发布。

主要更新如下:

重新设计的 Android Market 手势；支持 CDMA 网络；文字转语音系统(Text-to-Speech)；快速搜索框；全新的拍照接口；查看应用程序耗电；支持虚拟私人网络(VPN)；支持更多的屏幕分辨率；支持 OpenCore2 媒体引擎；新增面向视觉或听觉困难人群的易用性插件。

(4) Android 2.0/2.0.1/2.1 Eclair（松饼）: 2009 年 10 月 26 日发布。

主要更新如下:

优化硬件速度；“Car Home”程序；支持更多的屏幕分辨率；改良的用户界面；新的浏览器的用户接口和支持 HTML5；新的联系人名单；更好的白色/黑色背景比率；改进 Google Maps 3.1.2；支持 Microsoft Exchange；支持内置相机闪光灯；支持数码变焦；改进的虚拟键盘；支持蓝牙 2.1；支持动态桌面的设计。

(5) Android 2.2/2.2.1 Froyo（冻酸奶）: 2010 年 5 月 20 日发布。

主要更新如下:

整体性能大幅度提升；3G 网络共享功能；Flash 的支持；App2sd 功能；全新的软件商店；更多的 Web 应用 API 接口的开发。

(6) Android 2.3.x Gingerbread（姜饼）: 2010 年 12 月 7 日发布。

主要更新如下:

增加了新的垃圾回收和优化处理事件；原生代码可直接存取输入和感应器事件、EGL/OpenGLES、OpenSL ES；新的管理窗口和生命周期框架；支持 VP8 和 WebM 视频格式，提供 AAC 和 AMR 宽频编码，提供了新的音频效果器；支持前置摄像头、SIP/VOIP 和 NFC（近场通信）；界面简化、速度提升；更快更直观的文字输入；一键文字选择、复制和粘贴；改进的电源管理系统；新的应用管理方式。

(7) Android 3.0 Honeycomb（蜂巢）: 2011 年 2 月 2 日发布。

主要更新如下:

全新设计的 UI 增强网页浏览功能；n-app purchases 功能。

(8) Android 3.1 Honeycomb（蜂巢）: 2011 年 5 月 11 日发布。

主要更新如下:

经过优化的 Gmail 电子邮箱；全面支持 Google Maps；将 Android 手机系统跟平板系

统再次合并，从而方便开发者；任务管理器可滚动，支持 USB 输入设备（键盘、鼠标等）；支持 Google TV；可以支持 XBOX 360 无线手柄；Widget 支持的变化，能更加容易地定制屏幕 Widget 插件。

(9) Android 3.2 Honeycomb (蜂巢)：2011 年 7 月 13 日发布。

主要更新如下：

支持 7 英寸设备；引入了应用显示缩放功能。

(10) Android 4.0 Ice Cream Sandwich (冰激凌三明治)：2011 年 10 月 19 日在香港发布。

(11) Android 2.0 版本。

主要更新如下：

全新的 UI；全新的 Chrome Lite 浏览器，有离线阅读、16 标签页、隐身浏览模式等；截图功能；更强大的图片编辑功能；自带照片应用堪比 Instagram，可以加滤镜、相框，进行 360°全景拍摄，照片还能根据地点来排序；Gmail 加入手势、离线搜索功能，UI 更强大；新功能 People：以联系人照片为核心，界面偏重滑动而非点击，集成了 Twitter、Linkedin、Google+ 等通讯工具；有望支持用户自定义添加第三方服务；新增流量管理工具，可具体查看每个应用产生的流量，限制使用流量，到达设置标准后自动断开网络。

(12) Android 4.1 Jelly Bean (果冻豆)：2012 年 6 月 28 日发布。

新特性：

更快、更流畅、更灵敏；特效动画的帧速提高至 60fps，增加了三倍缓冲；增强通知栏；全新搜索；搜索将会带来全新的 UI、智能语音搜索和 Google Now 三项新功能；桌面插件自动调整大小；加强无障碍操作；语言和输入法扩展；新的输入类型和功能；新的连接类型。

(13) Android 4.2 Jelly Bean (果冻豆)：2012 年 10 月 30 日发布。

Android 4.2 沿用“果冻豆”这一名称，以反映这种最新操作系统与 Android 4.1 的相似性，但 Android 4.2 推出了一些重大的新特性，具体如下：

Photo Sphere 全景拍照功能；键盘手势输入功能；改进锁屏功能，包括锁屏状态下支持桌面挂件和直接打开照相功能等；可扩展通知，允许用户直接打开应用；Gmail 邮件可缩放显示；Daydream 屏幕保护程序；用户连点三次可放大整个显示屏，还可用两根手指进行旋转和缩放显示；专为盲人用户设计的语音输出和手势模式导航功能等；支持 Miracast 无线显示共享功能；Google Now 现可允许用户使用 Gmail 作为新的数据来源，如改进后的航班追踪功能、酒店和餐厅预订功能以及音乐和电影推荐功能等。

(14) Android 5.0 Key Lime Pie (青柠派)：待定。

2012 年中国智能手机市场不同操作系统分布情况如图 1.2 所示。

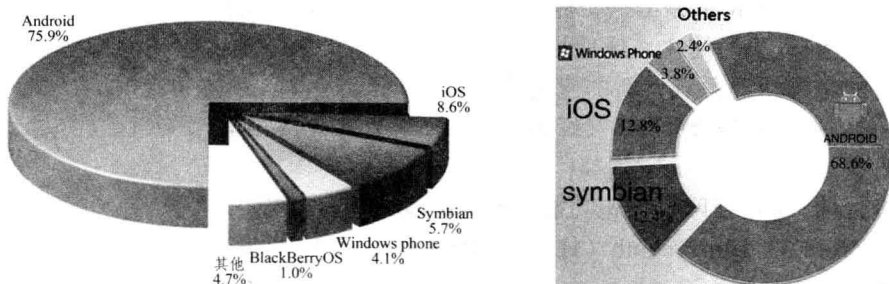


图 1.2 中国智能手机市场不同操作系统分布情况

1.1.2 Android API

API (Application Programming Interface, 应用程序编程接口) 其实就是操作系统留给应用程序的一个调用接口, 应用程序通过调用操作系统的 API 而使操作系统去执行应用程序的命令 (动作)。开发人员基于某软件或硬件访问 API 的一组例程, 而又无需访问源码, 或理解内部工作机制的细节。

(1) Android 平台可选 API。

Android 适用于各种各样的手机, 从最低端直到最高端的智能手机。核心的 Android API 在每部手机上都可使用, 但仍然有一些 API 接口有一些特别的适用范围, 这就是所谓的“可选 API”。这些 API 之所以是“可选的”, 主要是因为一个手持设备并不一定要完全支持这类 API, 甚至于完全不支持。例如, 一个手持设备可能没有 GPS 或 Wi-Fi 的硬件, 在这种条件下, 这类功能的 API 仍然存在, 但不会以相同的方式来工作。

(2) 最底层到最高层 API 核心包。

Android SDK 中的 API, 由最底层到最高层的核心包的主要作用如表 1.1 所示。

表 1.1 Android API 核心包的作用

命名空间	作 用
android.util	包含一些底层辅助类, 例如: 特定的容器类, XML 辅助工具类等
android.os	提供基本的操作服务, 消息传递和进程间通信 IPC
android.graphics	作为图形渲染包, 提供图形渲染功能
android.text	提供一套丰富的文本处理工作, 支持富文本、输入模式等
android.database	包含数据库相关的底层 API
android.content	包含获得以及发布数据的类, 提供大量的存储在设备上数据的服务
android.view	核心用户界面框架
android.widget	提供标准用户界面元素, 是组成我们界面的基本元素
android.app	提供高层应用程序模型, 实现使用 Activity
android.provider	提供方便调用系统提供的 content providers 的接口
android.telephony	提供 API 和手机设备的通话接口
android.webkit	包含一系列工作在基于 Web 内容的 API

1.2 Android 操作系统

1.2.1 Android 系统架构

Android 系统架构分为四层结构。从上层到下层分别是应用程序层、应用程序框架层、系统运行库层以及 Linux 内核层。Android 系统架构如图 1.3 所示。

1. 应用程序层

Android 平台不仅仅是操作系统, 也包含了许多应用程序。诸如 SMS 短信平台、电话拨号程序、图片浏览器、Web 浏览器等应用程序。本层的所有应用程序都是由 Java 编写的。

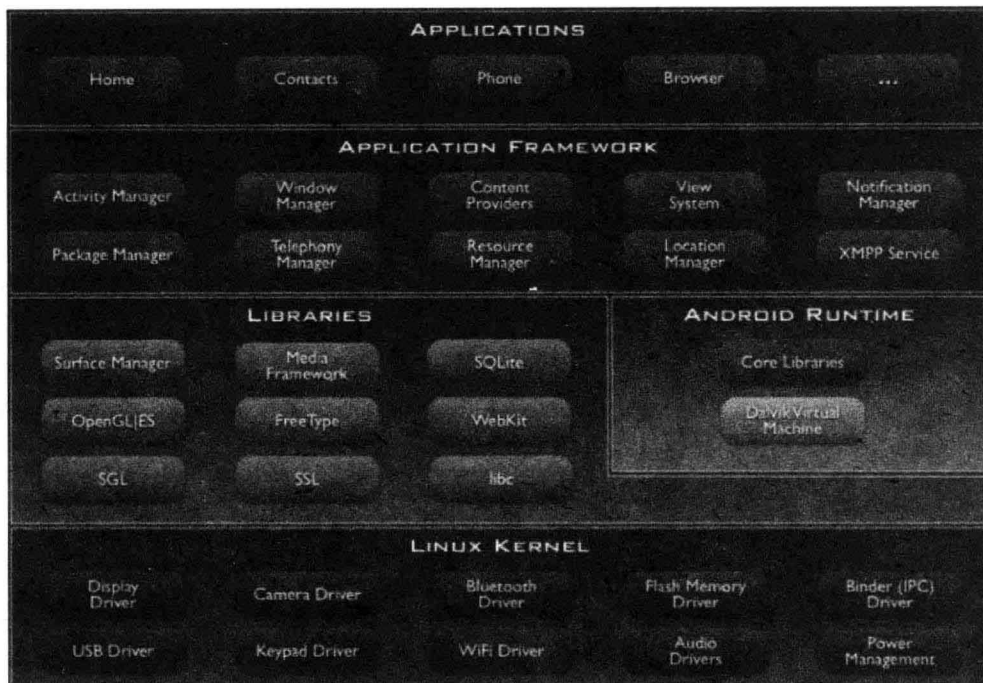


图 1.3 Android 系统架构

一般情况下，很多应用程序都是在同一系列的核心应用程序包中一起发布的。该层的应用程序是完全平等的，开发人员可以任意将 Android 自带的程序替换成自己的应用程序。

2. 应用程序框架层

对开发人员来说，接触最多的就是应用程序框架层了。该应用程序的框架设计简化了组件的重用，其中任何一个应用程序都可以发布自身的功能供其他应用程序调用，这也使用户可以很方便地替换程序的组件而不影响其他模块的使用。当然，这种替换需要遵循框架的安全性限制。

(1) Activity Manager (活动管理器)。

管理各个应用程序生命周期以及通常的导航回退功能。

(2) Window Manager (窗口管理器)。

管理所有窗口程序。

(3) Content Provider (内容提供器)。

使不同应用程序之间存取或者分享数据。

(4) View System (视图系统)。

构建应用程序的基本组件。

(5) Notification Manager (通告管理器)。

使得应用程序可以在状态栏中显示自定义的提示信息。

(6) Package Manager (包管理器)。

Android 系统内的程序管理。

(7) Telephone Manager (电话管理器)。

管理所有的移动设备功能。

(8) Resource Manager (资源管理器)。

提供应用程序使用的各种非代码资源,如本地化字符串、图片、布局文件、颜色文件等。

(9) Location Manager (位置管理器)。

提供位置服务。

(10) XMPP Service (XMPP 服务)。

提供 Google Talk 服务。

3. 系统运行库

该层包含两部分:分别是系统库和 Android 运行时库。系统库为一些 C/C++库,这些库能够被 Android 系统中不同的应用程序调用,并通过应用程序框架为开发者提供服务。而 Android 运行时库包含了 Java 编程语言核心库的大部分功能,提供了程序运行时所需调用的功能函数。

4. Linux 内核

Android 平台中操作系统采用的是 Linux 2.6 内核,其安全性、内存管理、进程管理、网络协议栈和驱动模型等基本依赖于 Linux。对于程序开发人员而言,该层为软件与硬件之间增加了一层抽象层,使开发过程中不必时时考虑底层硬件的细节。而对于手机开发商而言,对此层进行相应的修改即可将 Android 平台运行到自己的硬件平台之上。

1.2.2 Android 操作系统的四大应用程序组件

在 Android 中,一个应用程序可以使用其他应用程序的组件,这是 Android 系统一个非常重要的特性。在需要的时候 Android 会启动另外一个程序的部分代码,这部分代码实现了用户请求执行的动作的功能,而不是在用户的应用程序中直接包含另外一个应用程序的代码。为了实现这样的功能,Android 系统必须能够在其他应用程序请求的时候启动一个应用程序进程并实例化部分 Java 对象。因此,Android 应用程序包含一些运行应用程序所必需的能够被 Android 系统实例化的组件。

Android 架构里定义了四种应用程序组件,用户的 Android 应用程序都是由这四种组件(活动,服务,广播和意图接收器,内容提供者)组成的。这些组件主要包括以下四种类型:

(1) Activity (活动)。

一个 Activity 通常展现为一个可视化的用户界面。每个 Activity 都会有一个用于绘制用户界面的窗口。通常这样一个窗口会填充整个屏幕,当然这个窗口也可以比屏幕小并漂浮在其他窗口之上。展示 Activity 窗口的可视化内容区域是一些具有层次关系的视图,而视图则是由类 View 的子类表示的。视图就是 Activity 与用户交互的接口。例如,一个显示图片的视图,当用户单击的时候它可能会启动一个动作。Android 系统有许多开发人员可以直接使用的视图,包括按钮、文本域、滚动条、菜单、复选框等。

通过调用 Activity setContentView () 方法来设置展现 Activity 的窗口的视图。内容视图则是视图层次结构中的根节点视图。

Android 系统中的活动与桌面操作系统中独立的应用程序(比如办公软件等)类似。活动都是由一些可执行代码组成的,活动可以由用户或操作系统在需要时启动。活动可以与

用户进行交互，还可以通过查询（queries）和意图（intents）向其他活动或者服务（services）获取数据和服务。

用户编写的大部分可执行代码都是以活动的形式存在的。活动通常要负责屏幕的显示：每一个活动都有一个显示界面。如果一个活动不再处于活动期，操作系统可以通过清除它来提高手机内存的使用效率。

（2）Service（服务）。

Service 没有用户界面，但它会在后台一直运行。例如，Service 可能在用户处理其他事情的时候播放背景音乐，或者从网络上获取数据，或者执行一些运算，并把运算结果提供给 Activity 展示给用户。每个 Service 都扩展自类 Service。

Android 系统中的服务与桌面系统或者服务器中的后台程序有些类似。服务同样是由一些可执行代码构成的，它们启动以后就会一直在后台运行，直到手持设备关机为止。服务通常不会显示用户界面。应用程序可以连接到正在运行中的 Service，当连接到一个 Service 后，可以使用这个 Service 向外暴露的接口与这个 Service 进行通信。如果是播放音乐的 Service，这个接口可能允许用户暂停、停止或重新播放音乐。

与 Activity 以及其他组件一样，Service 同样运行在应用程序进程的主线程中，因此它们不能阻塞其他组件或用户界面，通常需要为这些 Service 派生一个线程执行耗时的任务。

（3）Broadcast Receiver。

Broadcast Receiver 不执行任何任务，仅仅是接受并响应广播通知的一类组件。大部分广播通知是由系统产生的。例如改变时区、电池电量低、用户选择了一幅图片或者用户改变了语言首选项。应用程序同样也可以发送广播通知，例如通知其他应用程序某些数据已经被下载到设备上可以使用了。

Broadcast Receiver 不包含任何用户界面。然而它们可以启动一个 Activity 以响应接收到的信息，或者通过 Notification Manager 通知用户。可以通过多种方式使用户知道有新的通知产生：闪动背景灯、振动设备、发出声音等。通常程序会在状态栏上放置一个持久的图标，用户可以打开这个图标并读取通知信息。

（4）Content Provider（内容提供者）。

应用程序可以通过 Content Provider 访问其他应用程序的一些私有数据，这是 Android 提供的一张标准的共享数据的机制。共享的数据可以是存储在文件系统中、SQLite 数据库中或其他一些媒体中。Content Provider 扩展自 ContentProvider 类，通过实现此类的一组标准的接口可以使其他应用程序存取由它控制的数据。然而应用程序并不会直接调用 ContentProvider 中的方法，而是通过类 ContentResolver。ContentResolver 能够与任何一个 ContentProvider 通信，它与 ContentProvider 合作管理进程间的通信。内容提供者通过使用标准的 URI（统一资源标识符）接口为其他应用程序提供数据服务，数据的请求者甚至不需要知道服务的提供者是谁。

任何时候当 Android 系统收到一个需要某个组件进行处理的请求的时候，Android 会确保处理此请求的组件的宿主进程是否已经在运行，如果没有，则立即启动这个进程，当请求的组件的宿主进程已经在运行，它会继续查看请求的组件是否可以使用，如果不能立即使用，它会创建一个请求的组件的实例来响应请求。

Android 操作系统中引入了意图（Intents）机制，使得程序脱离了对于某些软件的依赖。意图是一段可执行代码，负责响应其他活动或者服务发出的数据和服务请求。想要发送请

求的活动（客户端）首先需要创建一个意图对象，然后将这个意图对象提交给 Android 操作系统，最后由 Android 操作系统决定使用哪一个应用程序处理该意图。应用程序不是直接要求操作系统通过某种具体的软件发送一封 E-mail，而是将它要发送一封 E-mail 的“意图”告诉操作系统，由 Android 操作系统选择并启动 E-mail 发送程序，并最终完成任务。

在一个 Android 应用程序中可能不会同时用到这四种组件，但是对于一个优秀的应用程序，一定会积极使用系统提供的这些机制，而不是重新编写某项功能或者硬编码一些其他应用程序的引用。

1.3 Android 开发环境

编写 Android 程序，需要安装 Java JDK、Eclipse 和 Android SDK。

1.3.1 Java JDK 的下载及安装

开发 Android 应用程序需要安装 Java JDK。具体步骤如下：本节将带领读者逐步搭建自己的开发环境。

(1) 登录到官方网站 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 下载最新的 JDK 安装程序，如图 1.4 所示。

Java SE Development Kit 7u21		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM v6/v7 Soft Float ABI	65.09 MB	 jdk-7u21-linux-arm-sfp.tar.gz
Linux x86	80.35 MB	 jdk-7u21-linux-i586.rpm
Linux x86	93.06 MB	 jdk-7u21-linux-i586.tar.gz
Linux x64	81.43 MB	 jdk-7u21-linux-x64.rpm
Linux x64	91.81 MB	 jdk-7u21-linux-x64.tar.gz
Mac OS X x64	144.18 MB	 jdk-7u21-macosx-x64.dmg
Solaris x86 (SVR4 package)	135.84 MB	 jdk-7u21-solaris-i586.tar.Z
Solaris x86	92.08 MB	 jdk-7u21-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	22.67 MB	 jdk-7u21-solaris-x64.tar.Z
Solaris x64	15.02 MB	 jdk-7u21-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	136.09 MB	 jdk-7u21-solaris-sparc.tar.Z
Solaris SPARC	95.44 MB	 jdk-7u21-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	22.97 MB	 jdk-7u21-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.58 MB	 jdk-7u21-solaris-sparcv9.tar.gz
Windows x86	88.98 MB	 jdk-7u21-windows-i586.exe
Windows x64	90.57 MB	 jdk-7u21-windows-x64.exe

图 1.4 选择与机器系统相适应的 JDK

(2) 双击刚刚下载的 JDK 安装程序 `jdk-7u21-windows-i586.exe`, 根据提示将 JDK 安装到默认目录。

(3) 右键单击“我的电脑”, 依次选择“属性”/“高级”/“环境变量”, 在系统变量中新建一个名为 `JAVA_HOME`, 变量值为“`C:\Program Files\Java\jdk1.7.0_17`”(JDK 的安装路径)的环境变量。

(4) 新建 `CLASSPATH` 环境变量, 变量名为 `CLASSPATH`, 变量值为“`.;%JAVA_HOME%/lib/tools.jar; %JAVA_HOME%/lib/dt.jar;`”。

(5) 编辑 `PATH` 环境变量, 在“系统变量”列表框中, 选中 `PATH` 项, 单击下面的“编辑”按钮, 在“变量值”文本框的最前面加上“`%JAVA_HOME%/bin;`”。

(6) 安装完后检测。单击“开始”/“运行”, 输入 `cmd`, 在弹出的 DOS 窗口中输入 `javac` 并回车, 得到如图 1.5 所示结果, 说明配置成功。

```

C:\WINDOWS\system32\cmd.exe
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>javac
用法: javac <options> <source files>
其中, 可能的选项包括:
-g                生成所有调试信息
-g:none          不生成任何调试信息
-g:{lines,vars,source} 只生成某些调试信息
-nowarn          不生成任何警告
-verbose         输出有关编译器正在执行的操作的消息
-deprecation     输出使用已过时的 API 的源位置
-classpath <路径> 指定查找用户类文件和注释处理程序的位置
-cp <路径>       指定查找用户类文件和注释处理程序的位置
-sourcepath <路径> 指定查找输入源文件的位置
-bootclasspath <路径> 覆盖引导类文件的位置
-extdirs <目录>  覆盖所安装扩展的位置
-endorseddirs <目录> 覆盖签名的标准路径的位置
-proc:{none,only} 控制是否执行注释处理和/或编译。
-processor <class1>[,<class2>,<class3>... ] 要运行的注释处理程序的名称; 绕过默
认的搜索进程
-processorpath <路径> 指定查找注释处理程序的位置
-d <目录>           指定放置生成的类文件的位置
-s <目录>           指定放置生成的源文件的位置
-implicit:{none,class} 指定是否为隐式引用文件生成类文件
-encoding <编码>   指定源文件使用的字符编码
  
```

图 1.5 测试 JDK

1.3.2 Eclipse 集成开发环境的搭建

最新针对 Windows 的集成开发环境, Google 已经把 ADT 和 Eclipse 集成到一块了, 下载的时候根据自己使用的操作系统及 CPU 架构选择合适的版本下载。构建基于 Eclipse 的集成开发环境, 具体步骤如下:

(1) 下载安装集成开发环境。

到安卓官网: <http://developer.android.com/sdk/index.html>, 点击“Download the SDK ADT Bundles for Windows”按钮下载, 如图 1.6 所示。

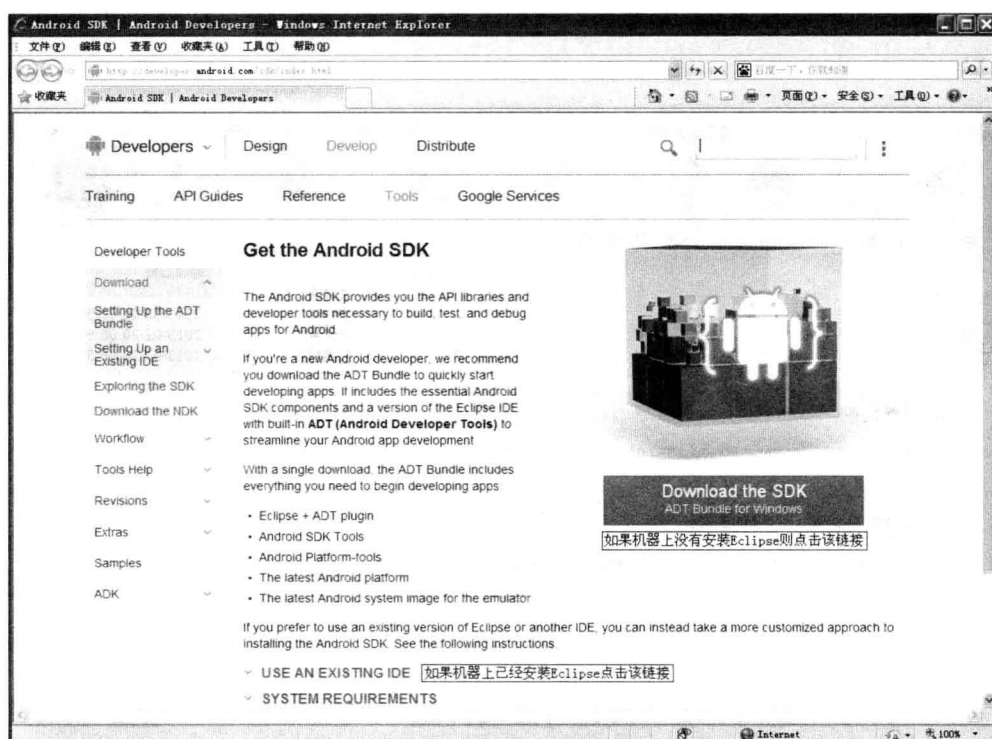


图 1.6 Android 官网下载链接

根据自己机器使用的操作系统及 CPU 架构选择合适的版本下载，如图 1.7 所示。



图 1.7 选择 Android SDK 安装版本

解压下载的“adt-bundle-windows-x86-20130219.zip”文件，里面包含两个目录：eclipse