

 高等教育规划教材

Android 移动应用 开发教程

刘佰龙 主编
谢红侠 赵莹 副主编



免费提供电子教案、源代码

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS



高等教育规划教材

Android 移动应用开发教程

刘佰龙 主编

谢红侠 赵莹 副主编

机械工业出版社

本书系统地介绍了 Android 移动应用开发技术,并根据课堂教学特点合理设计规模案例,既满足了学生的学习需求,也保证了课堂授课进度。本书首先介绍了 Android 开发的基础知识,包括发展历程、环境搭建、项目创建、Java 编程基础、UI 设计以及 Activity,然后讲解了 Android 中的数据存储,包括文件存储、SharedPreferences、SQLite 数据库和内容提供者等,接着介绍了 Android 的广播接收者和服务两大组件,随后介绍了 Android 的网络编程与 Internet 应用,最后以天气预报 APP 作为综合案例,展示了完整应用的开发过程。

本书深入浅出、通俗易懂、图文并茂。本书可作为普通高等院校计算机相关专业类课程教学用书,也可作为相关工作人员的自学参考用书。

本书配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册,审核通过后下载,或联系编辑索取(QQ: 2850823885, 电话: 010 - 88379739)。

图书在版编目 (CIP) 数据

Android 移动应用开发教程/刘佰龙主编. —北京:机械工业出版社, 2017.6

高等教育规划教材

ISBN 978-7-111-56926-8

I. ①A… II. ①刘… 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 (2017) 第 115697 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:郝建伟 责任编辑:郝建伟 范成欣

责任校对:张艳霞 责任印制:常天培

保定市中国画美凯印刷有限公司印刷

2017 年 6 月第 1 版·第 1 次

184mm × 260mm · 21 印张 · 512 千字

0001 - 3000 册

标准书号: ISBN 978-7-111-56926-8

定价: 55.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

服务咨询热线:(010)88379833

读者购书热线:(010)88379649

封面防伪标均为盗版

网络服务

机工官网:www.cmpbook.com

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

金书网:www.golden-book.com

出版说明

当前,我国正处在加快转变经济发展方式、推动产业转型升级的关键时期。为经济转型升级提供高层次人才是高等院校最重要的历史使命和战略任务之一。高等教育要培养基础性、学术型人才,但更重要的是加大力度培养多规格、多样化的应用型、复合型人才。

为顺应高等教育迅猛发展的趋势,配合高等院校的教学改革,满足高质量高校教材的迫切需求,机械工业出版社邀请了全国多所高等院校的专家、一线及教务部门教师,通过充分的调研和讨论,针对相关课程的特点,总结教学中的实践经验,组织出版了这套“高等教育规划教材”。

本套教材具有以下特点:

1) 符合高等院校各专业人才的培养目标及课程体系的设置,注重培养学生的应用能力,加大案例篇幅或实训内容,强调知识、能力与素质的综合训练。

2) 针对多数学生的学习特点,采用通俗易懂的方法讲解知识,逻辑性强、层次分明、叙述准确而精炼、图文并茂,使学生可以快速掌握,学以致用。

3) 凝结一线骨干教师的课程改革和教学研究成果,融合先进的教学理念,在教学内容和方法上做出创新。

4) 为了体现建设“立体化”精品教材的宗旨,本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、源代码或源程序、教学大纲、课程设计和毕业设计指导等资源。

5) 注重教材的实用性、通用性,适合各类高等院校、高等职业学校及相关院校的教学,也可作为各类培训班教材和自学用书。

欢迎教育界的专家和老师提出宝贵的意见和建议。衷心感谢广大教育工作者和读者的支持与帮助!

机械工业出版社

前 言

在当今大数据、“互联网+”等理念的影响下，移动网络平台的应用具有广泛的应用前景。而基于 Android 的移动开发所需成本低、上手快，使得该平台吸引了众多开发人员的加入。面对行业需求，当前各大高校也纷纷开设 Android 移动开发的相关课程。本书正是在这样一个背景下组织编写的。编写本书的目的是让广大教师能够在课堂有限的时间内完成 Android 开发的授课内容，让广大学生能够快速掌握 Android 编程。本书体现了计算机课程改革的方向之一。本课程建议授课学时为 32 学时，并要求先修 Java 编程技术课程。

本书从介绍 Android 的基础概念（起源、环境的搭建、项目创建和结构、Java 编程基础、UI 设计）开始，由浅入深，逐步介绍了 Android 移动应用开发中的四大组件（Activity、ContentProvider、BroadcastReceiver 和 Service），然后介绍了 Android 常用的数据存储方法，包括文件存储、SharedPreferences、SQLite 数据库以及 XML 序列化和解析，接着详细讲解了网络 and 多媒体编程的基础知识，最后以天气预报 APP 作为综合案例展示了 Android 程序开发的实现过程。本书每章都有相关案例，每个案例并不复杂，适合课堂教学，并能够有效帮助读者加深理解各个知识点。

本书所介绍的案例都是在 Eclipse 集成开发环境下编写并调试运行通过的。本书的第 12 章给出了一个完整的案例，以帮助读者顺利完成 Android 开发任务。本书每章的后面都附有实践与练习，每章都配有教学课件。

本书由中国矿业大学刘佰龙担任主编。具体编写分工如下：刘佰龙编写了第 1、4、10、11、12 章，赵莹编写了第 2、3 章，王冠军编写了第 5 章，张艳梅编写了第 6 章，刘晋编写了第 7 章，谢红侠编写了第 8、9 章。本书的统稿和校审由刘佰龙负责，本书样章的校审由谢红侠和赵莹协负责。非常感谢中国矿业大学计算机学院领导对出版本书的大力支持。

由于作者水平有限，书中难免存在不妥之处，请广大读者提出宝贵意见。

编 者

目 录

出版说明

前言

第1章 Android 基础知识	1	2.2.5 流程控制	38
1.1 Android 简介	1	2.3 面向对象的编程	43
1.1.1 Android 的起源	1	2.3.1 类和对象	43
1.1.2 Android 的发展历程	2	2.3.2 接口的定义与实现	46
1.1.3 Android 的体系结构	4	2.3.3 Java 中常用的类	47
1.1.4 Dalvik 虚拟机	5	2.4 小结	49
1.1.5 ART 模式	5	2.5 实践与练习	49
1.2 Android 开发环境的搭建	6	第3章 Android UI 设计	51
1.2.1 Windows 系统下的传统搭建方式	6	3.1 Android UI 简介	51
1.2.2 ADT Bundle 开发工具集合	13	3.2 Android 常用的 Widgets	53
1.2.3 Android Studio 开发环境	14	3.2.1 按钮	54
1.3 创建 Android 项目	16	3.2.2 静态文本框	55
1.3.1 Android 项目的创建步骤	16	3.2.3 编辑框	56
1.3.2 Android 项目的目录结构	17	3.2.4 单选按钮	57
1.3.3 Android 程序的打包	20	3.2.5 复选框	58
1.4 Android 项目工具	22	3.2.6 下拉列表框	59
1.4.1 Android 虚拟机	22	3.2.7 图像视图	61
1.4.2 虚拟机调试监控服务	23	3.3 Android 常用的布局管理器	62
1.4.3 Android 调试桥	24	3.3.1 布局文件的创建、修改与加载	62
1.5 Android 四大组件简介	26	3.3.2 线性布局	65
1.6 小结	27	3.3.3 相对布局	67
1.7 实践与练习	27	3.3.4 帧布局	71
第2章 Java 语言基础知识	29	3.3.5 表格布局	72
2.1 Java 语言概述	29	3.3.6 网格布局	75
2.1.1 Java 语言的发展史	29	3.3.7 绝对布局	76
2.1.2 Java 语言的特点	29	3.4 Android 屏幕中的单位	77
2.1.3 Java 的体系	30	3.5 Android 入门案例——电话 拨号器	78
2.2 Java 基本语法	31	3.6 Android 单击事件的处理方法	81
2.2.1 Java 程序的构成	31	3.7 样式和主题	83
2.2.2 数据类型、变量和常量	31	3.7.1 样式	83
2.2.3 运算符和表达式	34	3.7.2 主题	84
2.2.4 数组	37		

3.8 Android 程序的调试	85	4.6.4 singleTask 启动模式	125
3.8.1 Eclipse 平台调试	85	4.6.5 singleInstance 启动模式	126
3.8.2 Android SDK 调试	86	4.7 小结	127
3.8.3 JUnit 测试	88	4.8 实践与练习	127
3.9 小结	89	第5章 文件存取数据	129
3.10 实践与练习	90	5.1 数据存储概述	129
第4章 活动	92	5.2 Android 下的文件存储	130
4.1 活动概述	92	5.2.1 文件存储到手机内存	131
4.1.1 活动简介	92	5.2.2 文件存储到 SD 卡	132
4.1.2 活动的创建	92	5.2.3 案例——使用文件存储实现仿	
4.1.3 活动的启动和关闭	94	QQ 登录界面	133
4.1.4 案例——在工程中创建并启动		5.3 使用 SharedPreferences 存储	
一个新活动	95	数据	137
4.2 意图	98	5.3.1 案例——使用 SharedPreferences	
4.2.1 意图简介	98	实现仿 QQ 登录界面	139
4.2.2 意图的属性	98	5.4 XML 文件的序列化和解析	141
4.2.3 意图的投递	102	5.4.1 XML 文件	141
4.3 活动之间的跳转	103	5.4.2 XML 文件的序列化	142
4.3.1 显式跳转	103	5.4.3 案例——使用 XML 序列化存储仿	
4.3.2 隐式跳转	103	QQ 登录	143
4.3.3 案例——打开浏览器活动		5.4.4 XML 文件的解析	144
界面	103	5.4.5 案例——使用 PULL 解析仿	
4.3.4 案例——打开发送短信界面	105	QQ 登录界面	145
4.4 活动之间的数据传递	106	5.5 小结	146
4.4.1 使用 Intent 传递数据	106	5.6 实践与练习	146
4.4.2 使用 Bundle 传递数据	108	第6章 SQLite 数据库	148
4.4.3 案例——使用 Intent 和 Bundle		6.1 SQLite 数据库简介	148
传递数据	108	6.2 SQLite 数据库的管理	149
4.4.4 数据的回传	110	6.2.1 帮助类 SQLiteOpenHelper 及数据库的	
4.4.5 案例——头像选择	111	创建	149
4.5 活动的生命周期	115	6.2.2 SQL 语句操作数据库	151
4.5.1 认识活动生命周期	116	6.2.3 案例——使用 SQL 语句实现学生	
4.5.2 活动生命周期中的方法	116	信息管理	154
4.5.3 活动的 4 种状态	118	6.2.4 SQLite 操作 API	157
4.5.4 案例——生命周期演示	119	6.2.5 案例——使用 SQLite 操作 API	
4.6 活动的启动模式	122	实现学生信息管理	159
4.6.1 活动的任务栈	123	6.3 SQLite3 工具	161
4.6.2 Standard 启动模式	123	6.4 数据库的事务操作	162
4.6.3 singleTop 启动模式	125	6.5 数据库的可视化	163

6.5.1	ListView 控件	164	8.2.2	无序广播和有序广播	213
6.5.2	常用的数据适配器	165	8.3	自定义广播接收者	215
6.5.3	案例——使用 ArrayAdapter 配置 ListView 显示学生信息	170	8.3.1	自定义广播接收者的 实现步骤	215
6.5.4	案例——使用 SimpleAdapter 配置 ListView 显示学生信息	171	8.3.2	案例——商场优惠政策广播 实例	215
6.5.5	案例——使用自定义的适配器 配置 ListView 显示学生信息	174	8.4	预定义广播接收者	219
6.5.6	ListView 的单击事件和长按 事件	175	8.4.1	系统常用的广播接收者	219
6.6	小结	176	8.4.2	案例——IP 拨号器	221
6.7	实践与练习	177	8.4.3	案例——短信拦截器	224
第 7 章	内容提供者实现数据 共享	178	8.5	小结	225
7.1	内容提供者简介	178	8.6	实践与练习	225
7.2	通用资源标识符的使用	180	第 9 章	服务	227
7.2.1	通用资源标识符的组成	180	9.1	服务简介	227
7.2.2	通用资源标识符的解析	180	9.1.1	服务的概念	227
7.3	自定义内容提供者	181	9.1.2	服务的创建	227
7.3.1	内容提供者的创建	182	9.1.3	案例——锁屏和解锁广播在服务 中实现	228
7.3.2	访问内容提供者	183	9.2	服务的生命周期	229
7.3.3	案例——使用 ContentProvider 实现 学生信息的操作	184	9.3	开启服务的两种方式	231
7.4	预定义的内容提供者	192	9.3.1	标准方式开启和关闭服务	231
7.4.1	系统常用的内容提供者	192	9.3.2	案例——标准方式开启服务	231
7.4.2	案例——短信备份和还原	196	9.3.3	绑定方式开启和关闭服务	233
7.4.3	案例——联系人信息管理器	198	9.3.4	案例——绑定方式调用服务内部 方法	234
7.5	内容观察者	203	9.4	本地服务和远程服务	238
7.5.1	内容观察者简介	204	9.4.1	本地服务和远程服务的概念	238
7.5.2	案例——短信窃听器	206	9.4.2	案例——电话窃听器	239
7.6	小结	207	9.4.3	案例——自定义远程服务	242
7.7	实践与练习	207	9.5	小结	245
第 8 章	广播接收者	209	9.6	实践与练习	245
8.1	广播接收者简介	209	第 10 章	网络编程与 Internet 应用	247
8.1.1	广播接收者的创建与注册	209	10.1	网络编程基础	247
8.1.2	案例——简单的广播过程 演示	210	10.1.1	HTTP 简介	247
8.2	广播接收者的种类	212	10.1.2	Android 中的线程与消息处理 机制	248
8.2.1	常驻型广播和非常驻型广播	212	10.1.3	Tomcat 环境搭建	250
			10.2	通过 HTTP 访问网络	251
			10.2.1	向服务器请求和提交数据的	

方式	251	11.3.3 案例——开机自启动动画	296
10.2.2 HttpURLConnection 方式访问		11.4 播放音频	299
网络	253	11.4.1 使用 MediaPlayer 播放音频	300
10.2.3 案例——图片查看器	254	11.4.2 使用 SoundPool 播放音频	302
10.2.4 案例——仿 QQ 网络登录	257	11.4.3 案例——网络音频播放器	303
10.2.5 HttpClient 访问网络	262	11.5 播放视频	306
10.3 开源项目 Async HttpClient	264	11.5.1 使用 VideoView 控件播放	
10.3.1 使用 Async HttpClient 访问		视频	306
网络	264	11.5.2 使用 MediaPlayer 和 SurfaceView	
10.3.2 案例——文件上传	268	播放视频	309
10.4 文件的多线程下载	271	11.5.3 案例——简易视频播放器	309
10.4.1 多线程下载的原理	271	11.6 小结	311
10.4.2 案例——文件的多线程下载	272	11.7 实践与练习	311
10.5 小结	275	第 12 章 经典范例——天气预报	
10.6 实践与练习	275	APP	313
第 11 章 图形图像与多媒体开发	277	12.1 任务需求及准备工作	313
11.1 常用绘图类	277	12.2 UI 界面设计	315
11.1.1 Paint 类	277	12.3 响应代码实现	319
11.1.2 Canvas 类	279	12.3.1 设置城市功能模块	319
11.1.3 Bitmap 类	279	12.3.2 网络获取 XML 文件模块	320
11.1.4 BitmapFactory 类	280	12.3.3 天气业务类的构造	321
11.1.5 其他类	281	12.3.4 XML 文件的解析模块	322
11.1.6 案例——图片增加水印	282	12.3.5 刷新模块	324
11.2 图像特效处理	284	12.3.6 内容显示模块	324
11.3 Android 中的动画	289	12.4 APP 功能测试	327
11.3.1 补间动画	289	参考文献	328
11.3.2 逐帧动画	294		

第1章 Android 基础知识

随着智能手机的推广，Android 系统得到了较快的发展和广泛应用，越来越多的开发人员加入到 Android 开发的队伍当中。本章主要介绍 Android 的基础知识。本章要点如下。

- Android 的发展历程和体系结构。
- Android 开发环境（Windows + Eclipse）搭建。
- Android 项目创建、结构和调试方法。

1.1 Android 简介

下面介绍 Android 的起源、发展历程、体系结构等内容。

1.1.1 Android 的起源

Android 一词的本意指“机器人”，它最早出现于法国作家利尔亚当（Auguste Villiers de l'Isle - Adam）在 1886 年发表的科幻小说《未来夏娃》（L'ève future）中，作者将外表像人的机器起名为 Android。Android 可以简单理解为一种基于 Linux 的开放源代码的操作系统，主要应用于移动设备，如智能手机和平板电脑等。该操作系统由 Google 公司和开放手机联盟领导及开发。严格地讲，Android 是一个用于移动设备的软件栈，这个软件栈包含操作系统、一些中间件和关键的应用。Android 图标如图 1-1 所示。

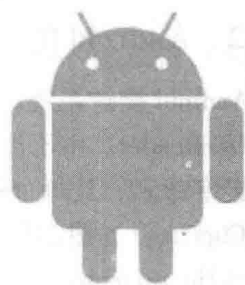


图 1-1 Android 图标

Android（安卓）操作系统最初产生于 Andy Rubin 创建的 Android 公司，当时它主要支持数码相机。2005 年 8 月，Google 注资并收购了该公司。2008 年 10 月，第一部 Android 智能手机发布。随后，Android 系统逐渐扩展到平板电脑及其他领域上，如电视、数码相机、游戏机等。2011 年第一季度，Android 在全球的市场份额首次超过塞班系统，跃居全球第一。2013 年第四季度，Android 平台手机的全球市场份额已经达到 78.1%。2013 年 9 月，全世界采用 Android 系统的设备数量已经达到 10 亿台。2014 年第一季度，Android 平台已占所有移动广告流量来源的 42.8%，首度超越 iOS。

Android 是一个对第三方软件完全开放的平台，开发者在为其开发程序时拥有更大的自由度，突破了 iPhone 等只能添加为数不多的固定软件的枷锁；与 Windows Phone 等厂商不同，Android 操作系统免费向开发人员提供，这样可节省成本。Android 平台的手机及平板电脑具有以下 5 大特点。

（1）开放性

Android 平台在优势方面首先就是其开放性，开放的平台允许任何移动终端厂商加入到 Android 联盟中。显著的开放性可以使其拥有更多的开发者。

(2) 挣脱运营商的束缚

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。例如，使用什么功能、接入哪种网络，几乎都受到运营商的控制。Android 上市后，用户可以更加方便地连接网络。

(3) 丰富的硬件选择

由于 Android 的开放性，众多的厂商会推出丰富多彩、各具特色的多种产品。功能上的差异和特色不会影响到数据同步、甚至软件的兼容性。

(4) 不受任何限制的开发商

Android 平台提供给第三方开发商一个十分宽泛、自由的环境，因此会诞生众多新颖别致的软件。但这也有其两面性，血腥、暴力、情色方面的程序和游戏如何控制是留给 Android 开发人员的难题之一。

(5) 无缝结合的 Google 应用

如今 Google 在互联网领域已经有了 10 多年的发展，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 Google 应用。

1.1.2 Android 的发展历程

Android 操作系统正式发布之前曾有两个预发布的内部测试版本，它们的代号分别是 Astro 和 Bender。由于涉及名称版权问题，从 2009 年 5 月开始，Android 的系统版本号以甜品的名字命名，且每个版本代表的甜品的尺寸越变越大，按照 26 个字母进行排序：纸杯蛋糕（Cupcake）、甜甜圈（Donut）、松饼（Eclair）、冻酸奶（Froyo）、姜饼（Gingerbread）、蜂巢（Honeycomb）、冰淇淋三明治（Ice Cream Sandwich）、果冻豆（Jelly Bean）、奇巧巧克力（KitKat）、棒棒糖（Lollipop）、棉花糖（Marshmallow）、牛轧糖（Nougat）。

下面详细介绍 Android 发展过程中的重要事件。

- 2003 年 10 月，Andy Rubin 等人创建 Android 公司，并组建 Android 团队。
- 2005 年 8 月，Google 收购 Android 及其团队并继续开发。Andy Rubin 则被任命为 Google 公司工程部副总裁，继续负责 Android 项目。
- 2007 年 11 月 5 日，Google 正式展示 Android 操作系统，并在当天宣布成立一个全球性的联盟组织——开放手持设备联盟（Open Handset Alliance, OHA）。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、软件开发商及电信运营商协作来共同研发改良 Android 系统。OHA 将支持 Google 发布的手机操作系统及应用软件，而 Google 公司则以 Apache 免费开源许可证的授权方式发布 Android 源代码。
- 2008 年 9 月，Google 正式发布 Android 1.0 系统——Astro。当月，全球第一台 Android 设备 HTC Dream（后更名为 T-Mobile G1）发布，并搭载该版本的操作系统。其主要特点如下：网页浏览器可以完全还原、显示 HTML 和 XHTML 的网页，并且可以通过多点触控对网页进行放大/缩小；可以通过 Android Market 下载应用程序，获得程序更新；支持多任务处理、即时通信，支持 Wi-Fi 和蓝牙；集成 Google 服务。
- 2009 年 2 月 2 日，Android 1.1（Bender）发布。该版本只被预装在 T-Mobile G1 上。

- 2009年4月，Google将Android系统升级至官方的Android 1.5版本（Cupcake，纸杯蛋糕），并开始以甜品命名Android版本号。
- 2009年9月，Android 1.6（Donut，甜甜圈）版本发布。该版本基于Linux 2.6.29内核，同时Google推出了搭载Android 1.6正式版的手机HTC Hero（G3）。
- 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开发主流分道扬镳。
- 2010年5月，Android 2.2系统发布，Google将该版本操作系统命名为Froyo（冻酸奶），在该版本中增加了对Flash的支持。
- 2010年10月，Android官方电子市场应用数突破10万个。
- 2010年12月，Android 2.3系统（Gingerbread，姜饼）发布，该版本基于Linux 2.6.35内核。
- 2011年2月，Google发布的Android 3.0系统（Honeycomb，蜂巢）专为平板电脑设计，并支持近场通信（NFC）。摩托罗拉公司于2011年2月24日发布的Motorola Xoom平板电脑是全球第一个使用该版本操作系统的设备。
- 2011年8月2日，Android手机已占据全球智能手机市场48%的份额，跃居全球第一。
- 2011年10月，Google发布Android 4.0系统（Ice Cream Sandwich，冰淇淋三明治），手机和平板电脑均可使用该版本操作系统。
- 2012年6月28日~2013年7月24日，Google分别推出了Android 4.1、Android 4.2、Android 4.3系统（Jelly Bean，果冻豆），主要是在Android 4.0基础之上进行了改善。
- 2013年11月1日，Android 4.4系统（KitKat，奇巧巧克力）正式发布。从功能上讲，Android 4.4提供了各种实用小功能，新的Android系统更智能，添加了更多的表情图案，UI的改进也更现代。但是，Google在该版本封锁了Flash Player，用户由Android 4.3升级到Android 4.4后无法播放Flash。
- 2014年6月25日于Google I/O 2014大会上发布Developer版（Android L），之后在2014年10月15日Android 5.0系统正式发布且名称定为Lollipop（棒棒糖）。
- 2015年5月28日，Google发布Android 6.0 Marshmallow（棉花糖）系统。
- 2016年8月22日，Google发布Android 7.0 Nougat（牛轧糖）系统。

图1-2所示为不同版本的Android代号及含义。

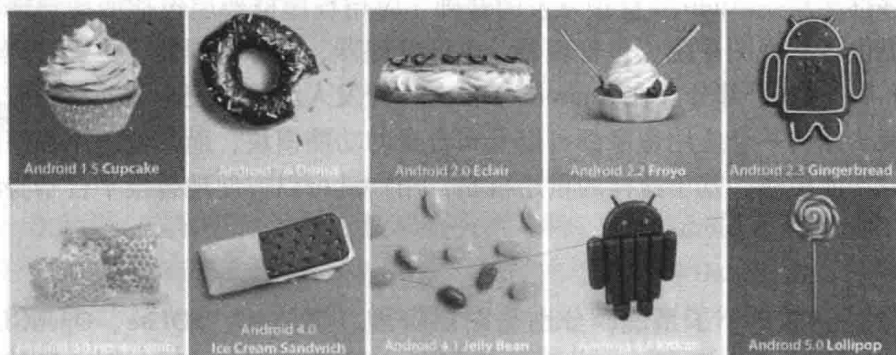


图1-2 Android历代版本代号及含义

1.1.3 Android 的体系结构

Android 是一个完整的操作系统，包含了中间件和一些关键的应用程序。在 Android 操作系统中共有 4 层架构，自上而下分别是应用程序层 (Applications)、应用程序框架层 (Application Framework)、系统运行库层 (Libraries) 和 Linux 内核层 (Linux Kernel)，如图 1-3 所示。

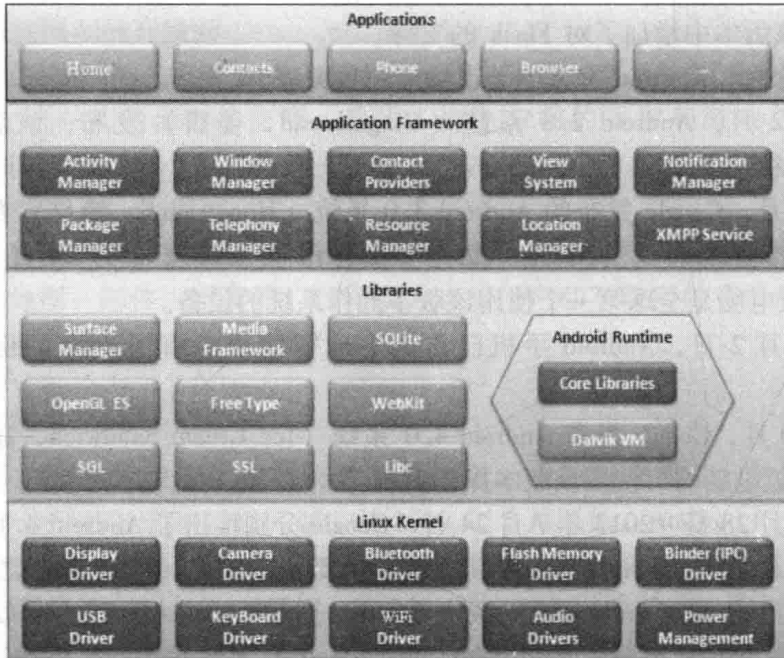


图 1-3 Android 体系结构框架

(1) 应用程序层

应用程序层提供了一些核心固有应用程序包，如电子邮件、短信、日历、地图、浏览器和联系人管理等。所有的应用程序都是用 Java 语言编写的。开发者可以编写属于自己的应用程序，而这些程序与那些核心应用程序彼此平等、友好共处。

(2) 应用程序框架层

应用程序框架层是 Android 应用开发的基础。应用程序框架层包括活动管理器、窗口管理器、内容提供者、视图系统、包管理器、电话管理器、资源管理器、位置管理器、通知管理器和 XMPP 服务 10 个部分。在 Android 平台上，开发人员可以完全访问核心应用程序所使用的 API 框架。任何一个应用程序都可以发布自身的功能模块，而其他应用程序则可以使用这些已发布的功能模块。基于这样的重用机制，用户就可以方便地替换平台本身的各种应用程序组件。

(3) 系统运行库层

系统运行库包括 9 个子系统，分别是图层管理、媒体库、SQLite、OpenGL ES、FreeType、WebKit、SGL、SSL 和 libc。Android 运行时 (RunTime) 包括 Android 的核心库和 Dalvik 虚拟机。前者包括 android.os、android.net、android.media 等，后者是一种基于寄存器的

Java 虚拟机，主要完成对生命周期的管理、堆栈的管理、线程的管理、安全和异常的管理以及垃圾回收等重要功能。

(4) Linux 内核层

Android 的核心系统服务依赖于 Linux 2.6 内核，如安全管理、内存管理、进程管理、网络协议栈、驱动模型和电源管理。Linux 内核也同时作为硬件与软件栈之间的抽象层。驱动模型包括显示驱动、摄像头驱动、键盘驱动、WiFi 驱动、Audio 驱动、Flash 内存驱动、Binder (IPC) 驱动、电源管理等。

从上面的介绍可以看出：

1) Android 的系统架构采用分层架构的思想，架构清晰、层次分明、协同工作。

2) Android 的系统架构不仅使开发人员从宏观上认识了 Android 系统，同时也给他们的学习与实践指明了方向。若从事 Android 应用开发，则应该重点研究 Android 的应用程序框架层和应用程序层；若从事 Android 系统开发，则应该研究 Android 的系统库和 Android 运行时；若从事 Android 驱动开发，则应该研究 Android 的 Linux 内核。

1.1.4 Dalvik 虚拟机

Android 系统是开源的，因此 Android 系统势必会配置在不同硬件配置的移动设备上。由于 Android 系统是以 Linux 系统为底层构建的，因此 Google 为了降低应用的开发难度，在 Linux 底层之上构筑了一个名为 Dalvik 的虚拟机。可以说，虽然 Android 程序是用 Java 语言编写的，但是 Android 程序是运行在 Dalvik 虚拟机中的。Dalvik 是 Google 公司设计用于 Android 平台的虚拟机，它不仅可以完成进程隔离和线程管理，而且可以提高内存的使用效率。

每个 Android 应用程序运行在自己的进程上，都有自己的 Dalvik 虚拟机实例。Dalvik 可以使一个设备能够有效地运行多个虚拟机。正是因为 Dalvik 虚拟机的存在，Android 系统的开发者只需使用 Google 提供的 SDK 即可较为轻松地按照一套“规则”创建 APP，不用顾忌硬件、驱动等问题。在每次执行应用时，Dalvik 虚拟机都会将程序的语言由高级语言编译为机器语言，这样当前设备才能够运行这一应用。

与基于栈的 Java 虚拟机不同，Dalvik 虚拟机是基于寄存器的，并通过一个 Java 语言编译器的编译来运行，而这个 Java 编译器已经被内置的“dx”工具转成 .dex 格式。DEX 是 Dalvik 虚拟机专用的文件格式。通常在一个 Java 应用中会定义很多类，编译完成后就会有相应很多的字节码文件 (CLASS 文件)，而在 CLASS 文件间会有不少冗余的信息；DEX 文件格式会把所有的 CLASS 文件内容整合到一个文件中，这样既减小了整体的文件大小，又提高了查找速度。

1.1.5 ART 模式

以往的 Android 系统为了兼容性，在 Dalvik 虚拟机编译时做出了妥协，即在每次运行应用时都需要一次编译，这样程序的执行效率会大大下降。自 Android 4.4 后，新增了一种应用运行模式——ART (Android Runtime) 模式。该模式与传统的 Dalvik 模式不同，ART 模式可以实现更为流畅的 Android 系统体验。事实上，Google 的这次优化源于其收购的一家名为 FlexyCore 的公司。该公司一直致力于 Android 系统的优化，ART 模式也是在该公司的优化方案上演进而来的。

ART 模式与 Dalvik 模式最大的不同在于：在启用 ART 模式后，系统在安装 APP 时会进行一次预编译，即会先将代码转换为机器语言存储在本地，这样在运行程序时就不会每次都进行编译了，大大提升了执行效率。

从编译过程来看，ART 模式确实能够改善 Android 平台一直以来在兼容性方面的问题，但 APP 经过预编译后的容量，以及是否兼容该模式也是需要重点考虑的问题。进入 ART 模式，首先需要进入手机的“开发者选项”中，在“选择运行环境”选项下选择两种模式的切换，如图 1-4 所示。需要注意的是，如果已经安装的 APP 不支持 ART 模式，则此时会无法正常使用该 APP。



图 1-4 ART 模式切换

1.2 Android 开发环境的搭建

目前，开发安卓 APP 有以下两类开发环境：Eclipse 和 Android Studio，所支持的计算机操作系统有 Windows、Linux 和 Mac OS。本节主要介绍在 Windows 操作系统下搭建 Android 开发环境的不同方式。

1.2.1 Windows 系统下的传统搭建方式

在 Windows 操作系统中搭建 Android 环境多数结合 Eclipse 开发平台。因此，本节主要介绍在 Eclipse IDE (Integrated Development Environment, 集成开发环境) 下搭建 Android 开发环境的传统步骤。

1. 安装 JDK

JDK (JavaSE Development Kit) 是 Java 开发工具箱。它是 Java 的核心，包含了 Java 的运

行环境 (Java Runtime Environment)、一系列 Java 工具和开发应用程序时调用的 Java 类库。进入 Oracle 公司的 JDK 下载页面, 选择操作系统的对应版本即可安装。安装结束后为了今后能够方便操作, 可以为 JDK 设置环境变量, 其步骤如下:

1) 用鼠标右键单击“我的电脑”, 在弹出的快捷菜单中单击“属性”命令, 选择“高级”选项卡, 单击“环境变量”按钮, 单击“系统变量”选项区中的“新建”按钮, 在弹出的对话框中添加以下环境变量。

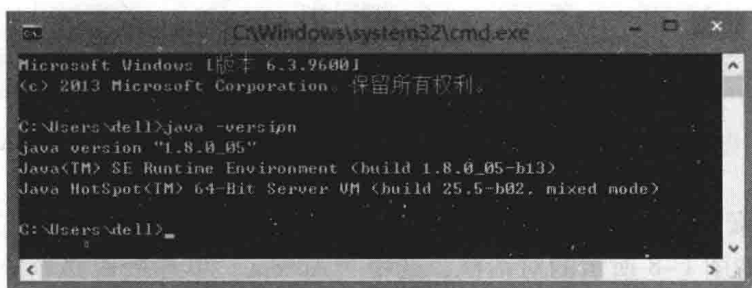
2) 新建变量名为“JAVA_HOME”, 变量值为“C:\Program Files\Java\jdk1.6.0_18”。

3) 新建变量名为“CLASSPATH”, 变量值为“.;%JAVA_HOME%\lib\tools.jar;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\bin”。

4) 双击“Path”变量: 在其值后面追加“;%JAVA_HOME%\bin”。

上述 4 步设置环境变量对 Android 开发环境的搭建不是必需的, 可以跳过。

为了检查 JDK 是否安装成功, 在设置完环境变量后, 打开 cmd 窗口, 输入“java -version”查看 JDK 的版本信息。如果出现如图 1-5 所示的结果, 则表示 JDK 安装成功。



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation. 保留所有权利。

C:\Users\de11>java -version
java version "1.8.0_05"
Java(TM) SE Runtime Environment (build 1.8.0_05-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.5-b02, mixed mode)

C:\Users\de11>
```

图 1-5 验证 JDK 安装是否成功

2. 安装 Eclipse 开发平台

Eclipse 是 Java 应用程序及 Android 开发的 IDE。Eclipse 的版本有多个, 对应不同类型和位数的操作系统。需要注意的是, 为了保证软件兼容性, 32 位的 Eclipse 要对应 32 位的 JDK。Eclipse 不需要安装, 在“<http://www.eclipse.org/downloads/index.php>”下载相应版本的 Eclipse 后, 解压到需要安装的文件夹内, 并在打开时设置其工作空间即可。图 1-6 所示为 Eclipse 的初始化界面。



图 1-6 Eclipse 的初始化界面

3. 安装 Android SDK

通过 Eclipse 来开发 Android 应用程序, 首先需要配置 Android SDK。Android SDK 提供了开发 Android 应用程序所需的 API 库和构建、测试、调试 Android 应用程序所需的开发工具。配置 Android SDK 的步骤如下:

1) 在 Android Developers 网站下载 Android SDK 压缩包, 并将其解压到需要的路径, 如“C:\Android\android-sdk-windows”。

2) 在安装目录中运行 Android SDK Manager.exe, 弹出如图 1-7 所示的对话框。若列表框中没有出现可安装的 SDK 包, 则可单击“Tools”菜单, 选中“Options”选项, 在弹出的对话框中勾选“Force https://”。

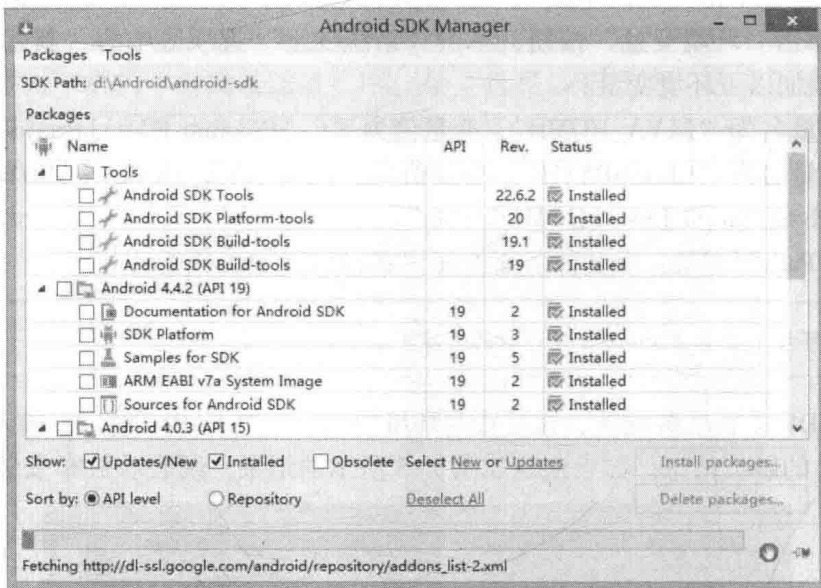


图 1-7 Android SDK Manager 对话框

3) 选择需要安装的 Android SDK 版本及相关文档或者相关包, 单击“Installation packages”按钮, 弹出如图 1-8 所示的对话框, 选中“Accept”单选按钮并单击“Install”按钮, 则开始下载安装所选的 Android SDK 包。

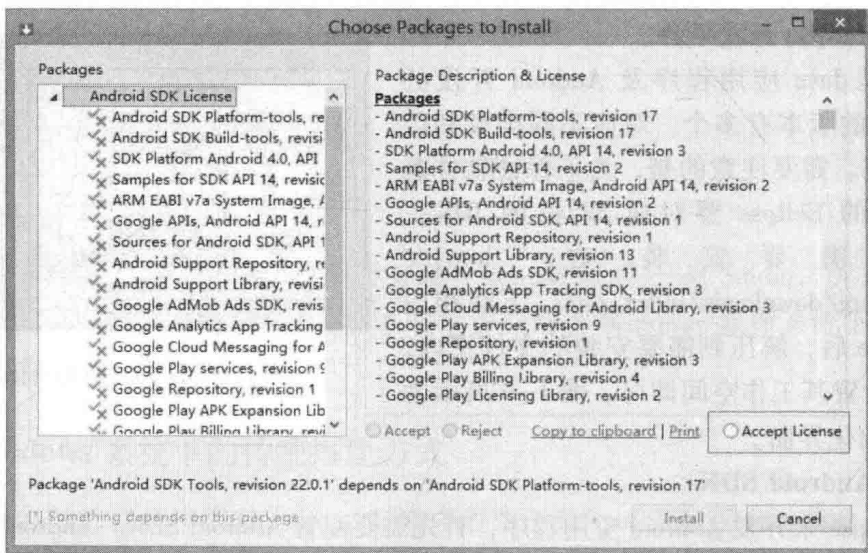


图 1-8 安装 SDK 对话框

4) 为了验证 Android SDK 是否安装成功, 可以在环境变量中的 PATH 值后追加 Android SDK 中的 tools 文件绝对路径 (如本机为“C:\Android\android - sdk - windows\tools”)。此时