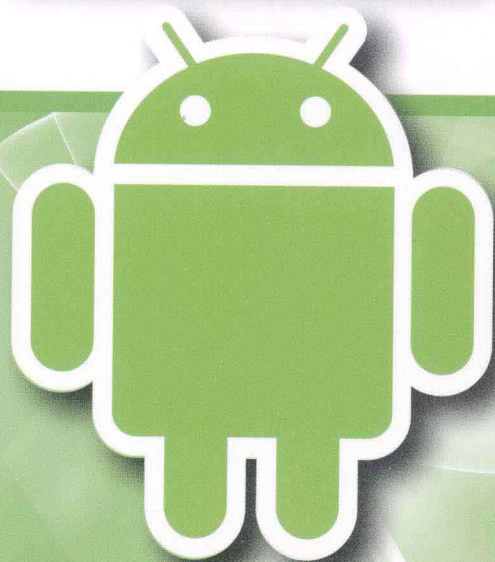


智能手机编程

智能手机编程课程组 编写



Android



中央广播电视大学出版社



智能手机编程

智能手机编程课程组 编写

中央广播电视大学出版社
北京

图书在版编目 (CIP) 数据

智能手机编程 / 智能手机编程课程组编写. —北京:
中央广播电视大学出版社, 2013. 8

ISBN 978 - 7 - 304 - 06266 - 8

I. ①智… II. ①智… III. ①移动电话机—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 176149 号

版权所有, 翻印必究。

智能手机编程

ZHINENG SHOUJI BIANCHENG

智能手机编程课程组 编写

出版·发行: 中央广播电视大学出版社

电话: 营销中心 010 - 58840200

总编室 010 - 68182524

网址: <http://www.crtvup.com.cn>

地址: 北京市海淀区西四环中路 45 号

邮编: 100039

经销: 新华书店北京发行所

策划编辑: 李永强

责任校对: 王 亚

责任编辑: 申 敏

责任印制: 赵联生

印刷: 北京密云胶印厂

印数: 0001-2000

版本: 2013 年 8 月第 1 版

2013 年 8 月第 1 次印刷

开本: 787 × 1092 1/16

印张: 10.5 字数: 230 千字

书号: ISBN 978 - 7 - 304 - 06266 - 8

定价: 20.00 元

(如有缺页或倒装, 本社负责退换)

前言 | PREFACE

【编写目的】

本书是一本专门介绍基于 Android 系统在移动智能终端上进行编程的应用技术教程，希望本书能够帮助读者开启进入移动智能终端编程的大门，成为目前移动互联网浪潮中的弄潮儿。

移动互联网正在快速崛起并深刻影响社会

从 2007 年苹果公司推出第一代 iPhone 智能手机开始，移动智能终端（智能手机、平板电脑）迅速普及开来，当今时代已经从互联网时代进入了移动互联网时代，从个人电脑时代进入了移动终端时代。市场研究公司 ABI Research 发布的数据显示，2013 年年底全球智能手机的保有量将达到 14 亿部，其中运行 Android 系统的手机占 57%，位列第一，我国智能手机的市场占有率将达到 55%。

由于智能手机拥有独立的操作系统，并搭载了多种功能，例如，全球定位系统、强大的摄影摄像功能、触摸式操作和简易快速的互联网访问等，因此手机已从单纯的通信工具演变成成为人们日常随身携带的工作、生活助手。

Android 应用软件开发市场广阔

庞大的智能手机市场蕴藏着巨大的机会和空间，搭载 Android 系统的智能手机在当前以及未来若干年内都将占据着这个规模较大的消费市场。Android 系统的开发商谷歌公司将 Android 系统开源，允许第三方对 Android 系统进行定制和二次开发；谷歌公司为推广 Android 系统，建立开发手机联盟，联络手机软硬件厂商、电信运营商和应用开发商等，共同推动 Android 智能手机的发展。根据最新的数据统计，Android 智能手机在国内市场已经占到将近八成的市场份额。

庞大的市场占有率，需要有丰富的应用软件来满足人们各种各样的需求。于是一个庞大的应用软件市场因此而出现，目前对 Android 应用开发人才的需求是巨大的，Android 软件

2 智能手机编程

工程师已经成为当前最为热门的职业之一。

本书将系统介绍 Android 系统开发环境的搭建和进行应用开发的相关基础知识，并结合典型案例来介绍 Android 系统中相关的知识和编程技巧，读者可以在学习案例的同时，了解智能手机编程的特点，为今后专业方面的学习打下坚实的基础。

【教材特点】

本教材为基于 Android 系统进行应用开发的学习用书，主要讲解基于 Android 系统进行各类应用开发的方法。本书在讲解 Android 应用开发的同时，穿插讲解了智能设备应用开发和程序设计中相关的知识，帮助读者在充分学习 Android 编程的同时，也能更多地学习和了解智能终端编程方面的知识。本书精心挑选了多个案例进行分析、设计和实现，在此过程中讲解 Android 相关的知识点，帮助读者快速学习、掌握和应用这些知识点。

本书与其他 Android 教材的不同之处在于，本书的章节是以案例驱动为导向的，在引导读者完成案例的分析、设计和实现之后，结合案例中使用到的知识点，进行相应的讲解和分析，在实践中学习知识点，在应用中掌握其用法。

在本书每章开始的“学习目标”部分，对通过本章的学习应该掌握的能力目标和知识目标提出了非常具体的要求，重点强调掌握和理解的学习内容。每一个案例都有相应的任务描述及功能分析、用户界面设计、步骤实现与核心代码设计和任务分析与总结，帮助读者一步一步地熟悉和掌握应用开发的步骤和流程。每章后还有主要术语及自测题，以达到帮助读者加深理解知识点和强化知识技能的目的。

【编者】

本书由美国莱斯龙系统有限公司高级工程师陈军峰和国家开放大学讲师王立共同编写完成，教学设计由国家开放大学王立完成。在本书的编写过程中得到了国家开放大学多位老师的大力支持，在此向他们表示衷心的感谢。

由于编者水平有限，错误与不足之处在所难免，恳请广大读者批评指正。

编者
2013年6月

目录 | CONTENTS

第 1 章 Android 概述	1
1.1 手机的演变	1
1.2 智能手机操作系统的现状	2
1.3 Android 操作系统	2
1.4 Android 的优势	3
1.5 Android 的系统架构	5
1.6 Android 的四个主要组件	6
1.7 Android 面临的挑战	8
本章小结	9
主要术语	10
自测题	12
第 2 章 Android 开发环境搭建	13
2.1 安装 Java 运行环境和 Eclipse 集成开发环境	13
2.1.1 下载和安装 JDK	13
2.1.2 下载安装 Android SDK ADT Bundle 软件包	17
2.2 Android SDK 版本的管理	19
2.3 创建虚拟设备	20
2.4 创建和运行 Android 应用	22
本章小结	26
主要术语	26
自测题	27
第 3 章 Android 项目结构和资源访问	28
3.1 Android 的项目结构	28
3.2 res 资源	30

3.3 通用的资源访问	31
3.3.1 在 Java 代码中使用资源	31
3.3.2 在 XML 中使用资源	32
3.4 使用字符串、颜色、尺寸资源	33
3.4.1 颜色值的定义	33
3.4.2 定义字符串、颜色和尺寸资源	34
3.4.3 在代码和 XML 配置文件中 使用资源信息	34
3.5 Drawable 资源	37
3.6 Android 项目布局文件	38
3.7 Android 项目配置文件	40
本章小结	42
主要术语	42
自测题	43
第 4 章 用户界面编程	44
4.1 简易计算器	44
4.1.1 任务描述	44
4.1.2 功能分析	45
4.1.3 用户界面设计	45
4.1.4 实现步骤与核心代码设计	52
4.1.5 任务分析与总结	58
4.2 创建 Activity	59
4.2.1 生成 Activity	59
4.2.2 界面布局文件	60
4.3 配置 Activity	61
4.4 启动、关闭 Activity	61
4.5 Activity 相互调用和数据交换	64
4.5.1 使用 Bundle 在 Activity 之间交换数据	64
4.5.2 启动其他 Activity 并返回结果	67
本章小结	70
主要术语	70
自测题	70
第 5 章 Android 的服务开发和广播事件	72
5.1 音乐播放器	72
5.1.1 任务描述	72

5.1.2	用户界面设计	73
5.1.3	功能分析	73
5.1.4	实现步骤与核心代码设计	74
5.1.5	任务分析与总结	80
5.2	Android 服务	80
5.2.1	Service 简介	80
5.2.2	创建服务的步骤	80
5.2.3	启动和停止服务	81
5.2.4	实例分析以及调用过程分析	81
5.2.5	Service 与 Thread 的区别	82
5.2.6	Android 中的系统服务	83
5.3	广播接收者	84
5.3.1	监听广播机制	84
5.3.2	监听广播的实现	85
5.3.3	广播事件生命周期	87
5.3.4	音乐播放器的广播事件	87
	本章小结	88
	主要术语	88
	自测题	89
第6章	Android 数据存储和数据共享	90
6.1	生词本	90
6.1.1	任务描述	90
6.1.2	用户界面设计	91
6.1.3	功能分析	92
6.1.4	实现步骤与核心代码设计	92
6.1.5	任务分析与总结	98
6.2	SQLite 数据库	99
6.2.1	SQLite 介绍	99
6.2.2	SQLiteOpenHelper 的使用方法	99
6.3	SQLiteDatabase 对象	100
6.4	SharedPreferences 的使用	101
6.4.1	SharedPreferences 简介	101
6.4.2	使用 SharedPreferences 存取数据	102
6.4.3	读写其他应用的 SharedPreferences	105
	本章小结	106

主要术语	106
自测题	107
第7章 Android 多媒体应用开发	109
7.1 视频播放器	109
7.1.1 任务描述	109
7.1.2 功能分析与设计	110
7.1.3 用户界面设计	110
7.1.4 实现步骤与核心代码设计	113
7.1.5 任务分析与总结	124
7.2 MediaPlayer 视频/音频播放	124
7.3 界面组件使用	125
7.3.1 ListView 简介	125
7.3.2 Looper 和 Handler 简介	126
本章小结	127
主要术语	127
自测题	128
第8章 地理位置服务应用开发	129
8.1 出行路线助手	130
8.1.1 任务描述	130
8.1.2 功能分析与设计	130
8.1.3 用户界面设计	131
8.1.4 实现步骤与核心代码设计	132
8.1.5 任务分析与总结	140
8.2 地图应用和地理位置服务	140
8.2.1 地图应用	140
8.2.2 地图服务厂商分析	140
8.2.3 基于地理位置的服务	141
8.3 Google Maps API 使用	142
8.3.1 Google Maps API 介绍	142
8.3.2 Map 的基本概念	142
8.4 Web App 或原生应用	144
本章小结	145
主要术语	146
自测题	146

参考文献	147
附 录	148
附录 1 Java 基础语法	148
附录 2 自测题部分参考答案	155

第 1 章 Android 概述

导 言

Android 是谷歌公司于 2007 年推出的基于 Linux 内核的开源操作系统，专门面向移动终端和智能设备；Android 为各个智能终端厂商提供了开放的合作策略，允许各大厂商基于各自需求深度定制 Android 系统。于是短时间内 Android 智能手机占领了过半的市场份额。Android 提供了一套完整的智能设备解决方案，涵盖了智能设备应用的方方面面。Android 系统更新频率较快，截至 2012 年年初已经发布了 11 个版本，目前最新的版本是 4.2。在 2013 年中将会发布最新的 Android 5.0 版本。本章将介绍智能手机的演变以及市场主流的智能手机操作系统；针对 Android 智能手机系统介绍相关的背景知识，并介绍 Android 系统的主要优势；同时，还会着重分析介绍 Android 系统的系统架构和四个主要组件。

学习目标

【掌握】

- Android 智能手机操作系统的系统架构。
- Android 应用框架的四个主要组件。

【理解】

- Android 智能手机操作系统的优势。

【了解】

- 当今三个主流智能手机操作系统的介绍。
- Android 智能手机操作系统的发展历程。
- Android 智能手机操作系统的现状。

1.1 手机的演变

手机从出现至今，经历了巨大的变化。从最早的“大哥大”到紧随其后的以诺基亚为代表的功能性手机，都曾经深深地影响了我们的生活，那个时代的手机主要是以通信为主，将拨打电话和收发短信以及简单的娱乐商务功能相结合，手机大都是以键盘操作为主。

2007 年美国苹果公司的 iPhone 智能手机横空出世，彻底改变了人们对于手机的理解。手机除了拨打电话和收发短信等基本功能之外，还可以提供类似于个人电脑的强大功能。用

户可以在手机上访问网络，利用全球定位系统（Global Positioning System, GPS）进行路线导航，使用影像拍摄系统随时随地摄影摄像，还可以玩游戏和阅读新闻；基于触摸屏的操作方式可以让人们非常方便地使用手机。这一年被认为是智能手机的初始之年。随着 iPhone 智能手机的广受欢迎，谷歌公司和微软公司相继推出了智能手机操作系统，智能手机在这些大厂商的推动下，迅速地普及开来。图 1-1 描述了手机的演变史，从第一代“大哥大”，到目前流行的智能手机。



图 1-1 手机演变史

1.2 智能手机操作系统的现状

目前主流的智能手机开发平台主要有三个：苹果公司的 iOS、谷歌公司的 Android 和微软公司的 Windows Phone。基于这三个智能手机平台的手机占据了市场的绝大部分份额，根据最新的数据统计，目前基于 Android 系统的智能手机占据了全球超过 70% 的份额，而在中国则超过了 80% 的市场份额。

相比之前的手机产品，智能手机除了可以拨打电话和收发短信之外，很多功能都可以和个人电脑相媲美。

- (1) 独立的操作系统，支持强大的多任务处理。
- (2) 简洁方便地访问互联网，访问速度大幅提高。
- (3) 物理键盘消失，被基于触摸屏的虚拟键盘取而代之，手机的操作方式也转变为触摸操作方式。
- (4) GPS 和移动互联网的结合。
- (5) 丰富强大的多媒体能力，摄像头可以拍摄高质量的照片和视频，并且可以以在线或离线的方式播放视频。
- (6) 丰富强大的应用商店，简洁、方便的查询和下载应用，良好的用户应用体验和性能。

1.3 Android 操作系统

Android 是谷歌公司开发的基于 Linux 的自由及开放源代码的智能手机操作系统（图 1-2），

它主要用于便携设备，如智能手机和平板电脑。Android 从 2008 年发布第一个版本开始，到现在已经发布了 11 个版本。谷歌公司使用糕点名称为不同的 Android 版本命名，其中最重要的版本有 Android 2.3 (Gingerbread, 姜饼), Android 3.0 (Honeycomb, 蜂巢), Android 4.0 (Ice Cream Sandwich, 冰淇淋三明治)。目前最新的 Android 版本是 4.2 (Jelly Bean, 果冻豆), 在 2013 年中将会发布更新的 Android 5.0 版本。

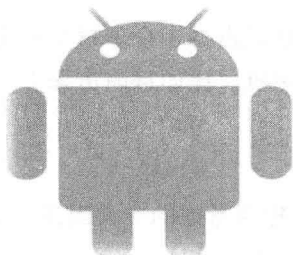


图 1-2 Android 系统的标识

与苹果的 iOS 和微软的 Windows Phone 相比, Android 平台是开放源代码的, 允许所有的公司和个人按照自己的意愿进行裁剪和定制。谷歌公司为 Android 智能手机平台建立了一个广泛的商业联盟推广平台, 同时也为 Android 智能手机平台的发展提供强大的后续支持, 这一系列的开发措施极大地激发了手机厂商的参与热情, 随后各个手机厂商纷纷推出 Android 平台的智能手机。

目前市场上主流的智能机厂商都推出了基于 Android 系统的智能手机, 其中韩国三星公司 (Samsung) 依靠 Android 智能手机已经成为当前全球较大的智能手机厂商。在国内, 以中兴 (ZTE)、华为 (Huawei) 为代表的传统通信厂商进入了 Android 智能手机市场, 还有小米 (XIAOMI)、魅族 (MEIZU) 等互联网公司, 依靠深度定制 Android 系统, 对用户使用界面进行优化, 提供比原有的 Android 系统更为简单易用的手机操作系统。

1.4 Android 的优势

目前市场上主流的智能机操作系统有 Windows Phone、iOS 和 Android, 每个系统背后都有一家实力雄厚的公司在支持, 每种操作系统都各有所长。Android 系统与其他两个系统相比有如下诸多的内在优势。

1. 开源的智能手机操作系统

所谓开源, 就是允许第三方查看系统的源代码, 并基于源代码进行二次开发, 从而定制符合自身硬件需求的系统。Android 开源的特性让手机厂商有更多优化和定制的空间, 对于开发者而言, 也可以对操作系统本身有更大的掌控余地。

2. 开发门槛低

Java 是目前最为流行的程序设计语言, Android 系统采用 Java 作为开发语言, 对于众多的开发者和厂商而言, 几乎不需要什么学习成本, 直接基于自己熟悉的 Java 为 Android 系统

开发应用。由于 Java 本身是跨平台的，所以基于 Java 的 Android 应用开发可以在所有的操作系统平台上进行。

iPhone 手机的 iOS 系统必须在苹果公司专有的 Mac 操作系统上才可以运行，而且需要重新学习一门新的开发语言：Objective-C。Windows Phone 手机需要基于 .NET 和 C#。这对应用的开发带来了或多或少的限制。

3. 市场份额高、用户群大

目前基于 Android 系统的手机已经占据了全球过半的市场份额，而在国内更是超过市场 80% 的份额是基于 Android 系统的手机。比如目前市场上热销的三星公司出品的 Galaxy S 系列手机。

4. Android 性价比占优

消费者选择产品，性价比是必然要考虑的一大因素，同等配置的手机，iPhone 手机的价格要比 Android 手机的价格高出一大截。相比而言，Android 平台的手机以其有竞争力的价格、用户体验和硬件配置，获得了大众的认可。

Android 平台简单实用，无论是功能还是外观设计，都可以与苹果的 iOS 一较高下。当消费者考虑价格因素之后，在数量众多的 Android 手机中，消费者总是会找到一款满意的 Android 手机取代价格高昂的 iPhone 手机。

5. 手机厂商的积极参与和鼎力支持

苹果的封闭式系统是它成功的一大法宝，从硬件到软件，再到其独特的推广方式，苹果已经形成了一个很好的、很完整的产业链。在一个封闭的圈子中创造了传奇产品，让消费者在使用体验在其产品中得到了很大的满足。但是随着乔布斯的离世，苹果如果不能继续保持创新力度，将无法使其长期占据有利位置。目前来看，苹果在创新上的优势日渐衰微，销量的增长速度与 Android 的爆发性增长之间的差异已经越来越明显。

现在世界智能手机厂家几乎都加入了 Android 的阵营，并推出了一系列的 Android 智能手机。三星、HTC、LG、华为、中兴、魅族以及小米等厂家都在力推 Android 系统的智能手机，基于 Windows Phone 系统的手机除诺基亚出货量较大之外，其余的厂商所占市场份额非常小，整个 Windows Phone 系统手机与 iPhone 手机或者 Android 系统手机的出货量相比，差距很大。Windows Phone 系统手机如要与其他两家三分天下，还需要走很长的一段路。

6. 庞大的应用商店和应用下载量

由于苹果的应用商店依托于其手机和平板电脑的拥有量，故它仍是目前最为赚钱的应用商店。但是 Android 的应用商店也正在迎头赶上，Android 在应用数量方面已经超过 iOS 平台。在 2012 年的第 4 季度，Android 应用商店 Google Play 收入增长了 200%，而苹果应用商店 App Store 收入仅增长 20%，目前 App Store 和 Google Play 收入的差距越来越小，按照这种趋势，App Store 的收入在 1 年到 2 年时间内会有被 Google Play 商店超越的可能。

2012 年 9 月谷歌公司表示，Android 设备应用下载次数达到 250 亿，苹果是在 2012 年 3 月份达到这一里程碑的。截至 2012 年 10 月份，苹果应用下载次数已突破 350 亿大关。Windows Phone 的应用商店中的应用目前为止还只有不到 20 万，在数量上和下载量上都与其

他两个应用商店相差甚远。

基于以上的分析，可以发现，基于 Android 智能手机进行应用开发，由于其开发的策略、良好的性价比、较低的开发门槛和庞大的用户群体，开发者可以专注于各自的业务领域，以较低的成本快速开发出满足用户需求的手手机应用，在移动互联网领域一展身手。

1.5 Android 的系统架构

Android 智能手机系统是基于 Linux 内核基础上搭建的全功能智能手机操作系统。本节将简要分析 Android 的系统架构，并针对开发应用时需要关注的内容，进行相应介绍。Android 系统架构如图 1-3 所示。

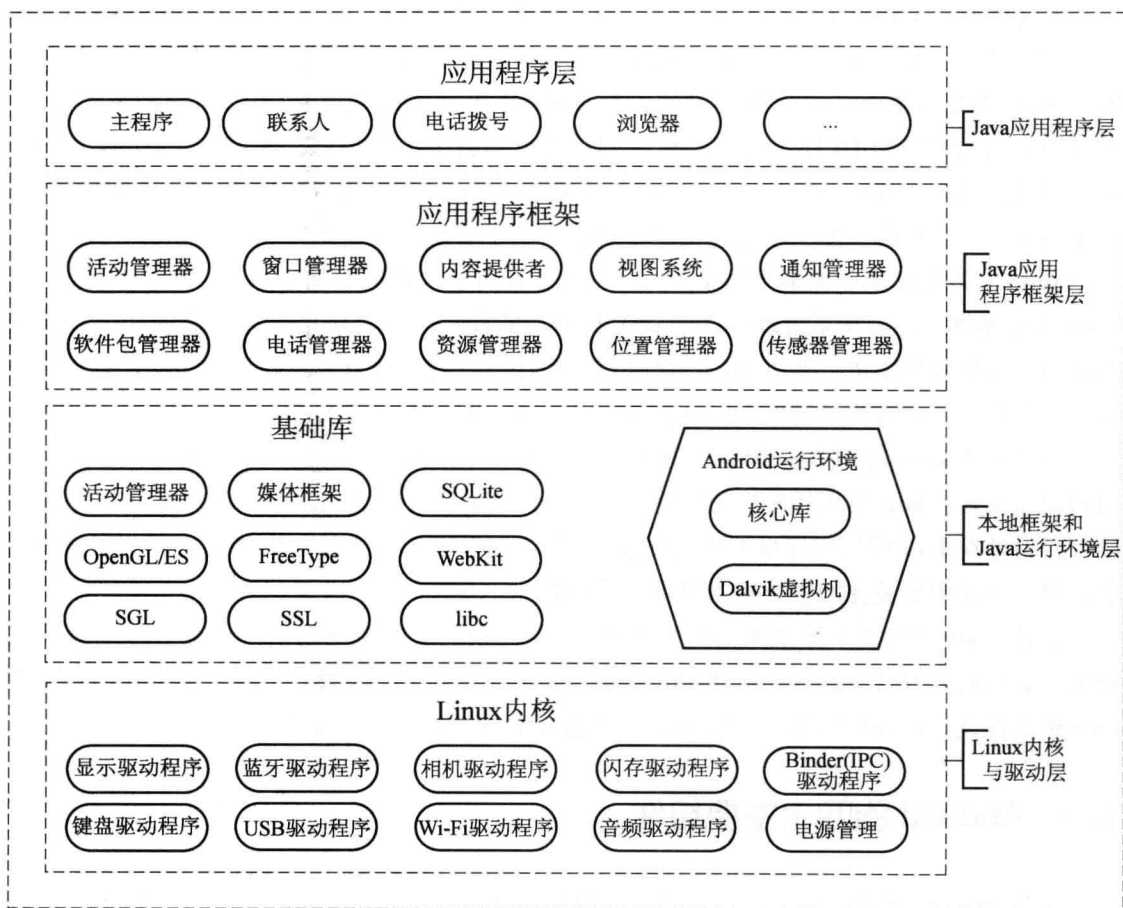


图 1-3 Android 系统架构

图 1-3 中描述了 Android 的系统架构，自上而下分成四个层次：Java 应用程序层 (Application)，Java 应用程序框架层 (Application Framework)，本地框架和 Java 运行环境层 (Libraries)，Linux 内核与驱动层。

Java 应用程序层主要为用户提供看得见的应用，智能手机用户通过触摸点击这些应用程

序来进行操作，比如拨打电话、浏览网页等。这类能够完成特定功能的应用程序基本都是通过 Java 语言编写的。Android 本身提供了桌面（Home）、联系人（Contacts）、拨打电话（Phone）、浏览器（Browsers）等很多基本的应用程序。绝大多数手机应用开发人员所做的事情都是在 Java 应用程序层上，开发实现特定功能的手机应用。

Java 应用程序框架层为开发者提供了调用 Android 基本功能和手机硬件系统的应用程序接口（Application Programming Interface, API），并实现了联系人、拨打电话、浏览器等常用功能，当然开发者也可以通过调用这些接口，直接制作自己的应用。由于应用开发与界面息息相关，故 Android 系统为开发者提供了很多开箱即用的界面组件，并在其官方网站上发布了大量的文字和视频培训资料，方便开发者学习和使用。开发者可以充分利用这些组件，专注于应用的业务，轻松地开发出优秀的 Android 手机应用。

本地框架和 Java 运行环境层是基于 Linux 内核开发的涉及 Android 系统架构底层的基础系统功能层，一般而言，绝大多数手机应用是无须直接访问这些底层系统功能层的。但由于 Linux 内核系统是使用 C 语言来开发的，而 Android 手机应用主要是以 Java 语言来实现的，所以 Linux 内核和应用程序层之间的交互调用需要通过 Dalvik 虚拟机来完成。Dalvik 是 Android 系统上 Java 的运行环境，它是谷歌公司基于 Java 语言规范开发出来的一个功能子集，遵循同样的语言标准，是一个经过精简定制之后用于满足移动设备的 Java 虚拟运行环境。

Android 系统构架中的最下层是 Linux 内核与驱动层，其服务基于 Linux 内核，它为 Android 系统提供了许多与系统硬件底层相关联的功能模块，主要包括安全管理模块、内存管理模块、进程管理模块、网络堆栈、硬件驱动程序等。这层涉及了与硬件的交互，对于手机应用开发而言，只需要了解即可，在实际的开发中一般是不会涉及的。

通过对 Android 系统架构的整体分析，可以发现，应用程序层包括手机平时接触到的应用和界面元素；Java 应用程序框架层则提供了 Android 系统提供的访问系统功能的应用程序接口，另外在 Java 应用程序框架中提供了丰富的开箱即用的用户界面组件。在实际的开发中，应用开发需要关注的主要就是这两层的内容。

了解 Android 的整体系统架构对于理解 Android 系统中的系统机制和应用开发有很大的帮助，能帮助开发者更好地使用 Android 提供的各项功能和系统服务，在开发中对出现的各种问题能有宽广的视野思路，这将是开发者能够做出优秀应用的技术基石。

1.6 Android 的四个主要组件

在对 Android 系统架构有了从上到下的整体认识之后，接下来将分析构成 Android 应用本身的主要组件。Android 应用程序都是由若干不同类型的组件组成的，但一个 Android 系统可以同时使用一个或多个组件，也可以不使用某种组件。应用组件是 Android 应用程序的重要基石，每个组件都可作为独立的实体存在，支持具体功能的实现。Android 中有四种不同类型的应用组件：Activity（活动）、Service（服务）、Broadcast Receiver（广播接收者）、Content Provider（内容提供者）。各种类型组件皆有各自特定的用途和生命周期，开发者需

要在使用过程中加以区别对待。

在介绍四个主要组件之前，首先需要介绍 Android 系统的 Intent（意图）。Intent 是 Android 系统中协助完成应用间的交互调用与通信的一种机制，负责在不同的组件之间传递消息，将一个组件的请求意图传给另一个组件。因此，Intent 包含具体请求信息的对象。针对不同的组件，Intent 所包含的消息内容也有所不同，不同组件的激活方式和传递的 Intent 类型也因组件不同而有所差异。Intent 不仅可用于应用程序之间，也可用于应用程序内部的 Activity 与 Service 之间的交互。因此，Intent 在不同的活动和组件之间起着媒体中介的作用，专门提供组件互相调用的相关信息，实现调用者与被调用者之间的解耦。在第 4 章用户界面编程中，还将对 Intent 进行详细介绍。

下面对这四个组件进行简要分析，感兴趣的读者可以访问 Android 的官方网站，里面提供了相应的拓展阅读内容。

1. Activity

从字面上理解，Activity 是活动的意思。在 Android 系统中，Activity 通常展现为可视化的用户界面，提供 Android 程序与用户交互的窗口，是 Android 组件中最基本也是最复杂的组件。从视觉效果来看，一个 Activity 占据当前的窗口，响应所有窗口事件，具备控件、菜单等界面元素。从内部逻辑来看，Activity 为保持各个界面的状态稳定，需要在保存数据和调用系统功能、妥善管理生命周期和实现界面之间设置跳转逻辑等。对于开发者而言，一般会创建 Activity 的子类，在其基础上定义界面布局、添加业务逻辑等。

2. Service

Service（服务）是运行在后台的一个组件，从某种意义上说，Service 就像没有界面的 Activity。它们在概念上比较接近，都封装有一个完整的功能逻辑实现，接受调用的请求，完成相应的操作，定义接收的 Intent，提供同步和异步操作的接口。Android 系统中，Service 一般用于执行长时间运行的且不需要提供用户界面的操作，例如，在后台下载文件、播放音乐等。在播放音乐的同时还可以做其他事情，且不会阻塞用户与其他 Activity 的交互。

3. Broadcast Receiver

Broadcast Receiver（广播接收者）是一种广泛运用在应用程序之间传输信息的机制，广播就是实现一个事件源被多个接收者接收。Broadcast Receiver 是对发送出来的广播进行过滤接收并响应的一类组件，通过它可以实现组件之间异步的消息通信。Broadcast Receiver 不包含任何用户界面，也不执行复杂的逻辑操作，但它们可以启动 Activity 以响应接收到的信息，或者通过 Notification Manager（通知管理）来通知用户。在用户界面上可以通过多种方式获取新的消息或通知，例如闪动背景灯、振动设备、发出声音等。

通常程序会在状态栏上放置图标，用户可以通过单击打开这个图标来读取通知信息。在 Android 中，Intent 是对程序中动作和行为的抽象描述，它负责在组件之间或程序之间进行消息传递；而 Broadcast Receiver 组件把 Intent 作为一个消息在组件间进行广播，所有对其感兴趣的组件程序都可以接收到该 Intent，然后执行相应操作。