

## 第 1 章 Android 基本概念

本章介绍 Android 基本概念方面的内容，包括 Android 平台特性、Android 系统架构、Android 开发框架以及 OMS 介绍。

### 1.1 Android 简介



图 1-1 Android 系统的 Logo

Android 一词的本意指“机器人”，它是由 Google 公司于 2007 年 11 月推出的基于 Linux 平台的开源手机操作系统，依靠 Google 的强大开发和媒体资源，Android 成为众多手机厂商竞相追逐的对象，逐渐发展成为目前最流行的手机开发平台。图 1-1 所示为 Android 系统的 Logo。

Android 是一个包括操作系统、中间件、用户界面和关键应用软件的移动设备软件堆。换句话说，Android 是基于 Java 并运行在 Linux 内核上的轻量级操作系统，功能全面，包括一系列 Google 公司在其内置的应用软件，如电话、短信等基本应用功能。图 1-2 所示为 Android 的模拟器，从中可以大概了解 Android 的运行界面。



图 1-2 Android 模拟器



问：模拟器和真机有什么区别呢？

答：

- 模拟器不支持呼叫和接听实际来电，但可以通过控制台模拟电话呼叫（呼入和呼出）。
- 模拟器不支持USB连接。
- 模拟器不支持相机/视频捕捉。
- 模拟器不支持音频输入（捕捉），但支持输出（重放）。
- 模拟器不支持扩展耳机。
- 模拟器不能确定连接状态。
- 模拟器不能确定电池电量水平和充电状态。
- 模拟器不能确定SD卡的插入/弹出。
- 模拟器不支持蓝牙。

## 1.2 Android 平台特性

随着科技的发展，移动电话（Mobile Phone）正向着智能化的方向迈步，并逐步成为多种工具的功能载体，而 Android 就是这样一个智能手机的平台，一个多种工具的功能载体。

### 1. 通信工具

移动电话的最基本功能即为通信，因此，使用运营商提供的通信网络进行语音通话也是 Android 平台的最基本功能。除了传统的语音通话功能外，Android 平台还具有短消息功能，以及通常移动电话都具有的个人信息系统管理方面的功能（如电话本等）。

### 2. 网络工具

随着数字业务使用的普遍化，移动电话通常作为网络工具使用，这样移动电话可以完成电脑的部分功能。由此，Android 平台在网络方面的功能主要包括浏览器、IM（即时信息）、邮件等，基本包含了网络方面的大部分功能。

### 3. 媒体播放器

随着多媒体技术的应用，在移动电话上进行音频和视频播放已经成为经常使用的功能。由此，Android 平台具有支持更多的音频/视频格式，支持更高分辨率的视频、更流畅地播放，以及和网络结合的流媒体方面等功能。

### 4. 媒体获取设备

随着移动电话与媒体获取设备的集成日益增强，Android 平台提供了照相机、录音机、摄像机等功能。



## 5. 多类型的连接设备

Android 平台提供了多种连接方式，如 USB、GPS、红外、蓝牙、无线局域网等。

## 6. 具有友好和绚丽的用户界面

Android 平台具有友好的用户界面，使用户容易学习和操作，同时具有绚丽的用户界面，具有良好的视觉效果。

## 7. 可以个性化定制的平台

Android 平台对于用户的个性化需求，提供了全面自定义手机的功能。

除了以上介绍 Android 平台的功能以外，其在技术上还具有以下几个方面的特性。

- ❑ 全开放智能移动电话平台。
- ❑ 支持多硬件平台。
- ❑ 使用众多的标准化技术。
- ❑ 核心技术完整、统一。
- ❑ 完善的SDK和文档。
- ❑ 完善的辅助开发工具。

## 1.3 Android 系统架构

如图 1-3 所示，Android 系统从下至上分为四层：Linux 内核、Android 核心库及 Android 运行时环境（Android Runtime）、应用程序框架以及应用程序等。



图 1-3 Android 系统框架



### 1.3.1 Linux 内核 (Linux Kernel)

Android 的核心系统服务依赖于 Linux 2.6, 例如, 安全、内存管理、进程管理、网络堆栈、驱动模型。Linux Kernel 也作为硬件和软件之间的抽象层, 它隐藏具体硬件细节而为上层提供统一的服务。

除了标准的 Linux 内核以外, Android 系统还增加了内核的驱动程序, 例如, 显示驱动、蓝牙驱动、相机驱动、闪存卡驱动、Binder IPC 驱动、输入设备驱动、USB 驱动、WiFi 驱动、音频系统驱动、电源管理等, 为 Android 系统的运行提供基础性支持。

这样分层的好处就是使用下层提供的服务而为上层提供统一的服务, 屏蔽本层及以下层的差异, 当本层及以下层发生了变化时, 不会影响到上层。也就是说, 各层各尽其职, 各层提供固定的 SAP (Service Access Point), 即高内聚、低耦合。

### 1.3.2 Android 核心库 (Libraries)

Android 包含一个 C/C++库的集合, 以供 Android 系统的各个组件使用。这些功能通过 Android 的应用程序框架 (Application Framework) 展现给开发者。下面列出一些核心库:

- ❑ 系统 C 库——由 BSD 继承衍生的标准 C 系统函数库 (libc), 调整为基于嵌入式 Linux 设备的库。
- ❑ 媒体库——基于 PacketVideo 的 OpenCORE。这些库支持播放和录制多种流行的音频和视频格式, 以及多种媒体格式的编码/解码格式, 包括 MPEG4、H.264、MP3、AAC、AMR、JPG、PNG。
- ❑ 界面管理——显示子系统的管理器, 管理访问显示子系统和无缝组合多个应用程序的二维和三维图形层。
- ❑ LibWebCore——新式的 Web 浏览器引擎, 驱动 Android 浏览器和可嵌入的 Web 视图。
- ❑ SGL——Skia 图形库, 基本的 2D 图形引擎。
- ❑ 3D libraries——基于 OpenGL ES APIs 的实现。该库使用硬件 3D 加速或使用高度优化的 3D 软加速。
- ❑ FreeType ——位图 (Bitmap) 和矢量 (Vector) 字体渲染。
- ❑ SQLite ——所有应用程序都可以使用的强大而轻量级的关系数据库引擎。

### 1.3.3 Android 运行时环境 (Android Runtime)

在 Linux 内核层上还有一个 Android 运行时层, 该层包括 Dalvik 虚拟机及 Java 核心库, 提供了 Java 编程语言核心库的大多数功能。

Dalvik 虚拟机是 Android 使用的 Java 虚拟机。每一个 Android 应用程序是 Dalvik 虚拟机中的实例, 运行在它们自己的进程中。Dalvik 虚拟机设计成在一个设备上可以高效地运行多个虚拟机。Dalvik 虚拟机可执行的文件格式是 .dex, .dex 格式是专为 Dalvik 设计的一种压缩格式, 适合内存和处理器速度有限的系统。



大多数虚拟机（包括 JVM）都是基于栈的，而 Dalvik 虚拟机则是基于寄存器的。两种架构各有优劣，一般而言，基于栈的机器需要更多的指令，而基于寄存器的机器指令更大。DX 是一套工具，可以将 Java .class 转换成 .dex 格式。一个 .dex 文件通常会有多个 .class。由于 .dex 有时必须进行最佳化，会使文件大小增加 1~4 倍，以 ODEX 结尾。

Dalvik 虚拟机依赖于 Linux 内核提供基本功能，如线程和底层内存管理。

### 1.3.4 Android 应用程序框架（Application Framework）

位于 Android 程序库和运行时上面的是应用程序框架层。通过提供开放的开发平台，Android 使开发者能够访问核心应用程序所使用的 API 框架，这样使得组件的重用得以简化，任何应用程序都能发布它的功能且任何其他应用程序可以使用这些功能（需要服从框架执行的安全限制）。从而使开发者可以编制极其丰富和新颖的应用程序，自由地利用设备硬件优势、访问位置信息、运行后台服务、设置闹钟、向状态栏添加通知等。

每个应用程序其实是一组服务和系统，包括：

- ❑ 视图（View）——丰富的、可扩展的视图集合，用来构建应用程序。包括列表（ListView）、网格（Grid）、文本框（EditText/TextView）、按钮（Button）等，甚至是可嵌入的网页浏览器（WebView）。
- ❑ 内容提供者（Content Providers）——使应用程序可以访问其他应用程序（如通讯录）的数据，或共享自己的数据。
- ❑ 资源管理器（Resource Manager）——提供对于非代码资源的访问，如本地化字符串、图形和布局文件。
- ❑ 通知管理器（Notification Manager）——使应用程序能够在状态栏显示自定义的提示信息。
- ❑ 活动管理器（Activity Manager）——管理应用程序生命周期，并提供常用的导航回退功能。

### 1.3.5 Android 应用程序

Android 装配一个核心应用程序集合，连同系统一起发布，这些应用程序包括电子邮件客户端、SMS 程序、日历、地图、浏览器、联系人和其他设置等。而所有应用程序都是用 Java 语言编写的，由用户开发的 Android 应用程序和 Android 核心应用程序是同一层次的。

## 1.4 Android 开发框架

Android 系统作为一个开放的系统，它体积庞大，对于不同的开发者来说，在开发过程中并不需要掌握整个 Android 系统，只需要进行其中某一部分的开发。由此，从功能上将





Android 开发分为移植开发移动电话系统、Android 应用程序开发，以及 Android 系统开发三种。

从商业模式的角度来讲，移植开发移动电话系统和 Android 应用程序开发是 Android 开发的主流。移植开发移动电话系统主要是由移动电话的制造者来进行开发，其产品主要是 Android 手机；而公司、个人和团体一般进行 Android 应用程序的开发，产生各种各样的 Android 应用程序。

对于 Android 移植开发，其主要工作集中于 Linux 内核中的相关设备驱动程序及 Android 本地框架中的硬件抽象层接口的开发；对于 Android 应用程序开发，其开发的应用程序与 Android 系统的第四个层次的应用程序是一个层次的内容；对于 Android 系统的开发，涉及 Android 系统的各个层次，一般情况下是从底层到上层的整体开发。

Android 开发框架包括基本的应用功能开发、数据存储、网络访问三大块。

### 1.4.1 应用方面

一般而言，一个标准的 Android 程序包括 Activity、Broadcast Intent Receiver、Service、Content Provider 四部分。

#### 1. Activity

Activity 是在 Android 应用开发中最频繁、最基本的模块。在 Android 中，Activity 类主要与界面资源文件相关联（res/layout 目录下的 xml 资源，也可以不含任何界面资源），包含控件的显示设计、界面交互设计、事件的响应设计以及数据处理设计、导航设计等 Application 设计的方方面面。

因此，对于一个 Activity 来说，它就是手机上的一个界面，相当于一个网页，所不同的是，每个 Activity 运行结束时都返回一个返回值，类似一个函数。Android 系统会自动记录从首页到其他页面的所有跳转记录并且自动将以前的 Activity 压入系统堆栈，用户可以通过编程的方式删除历史堆栈中的 Activity Instance。

#### 2. Broadcast Intent Receiver

Intent 为不同的 Activity 进行跳转提供了机制，譬如从 A Activity 跳转到 B Activity，使用 Intent 来实现，语句如下。

```
Intent in = new Intent(A.this, B.class);
startActivity(in);
```

Broadcast Intent Receiver 为各种不同的 Android 应用程序间进行进程间的通信提供了可能。如当电话呼叫来临时，可以通过 Broadcast Intent Receiver 发布广播消息。对用户而言，用户是无法看到 Broadcast Intent Receiver 事件的，它对用户是不透明的，Broadcast Intent Receiver 通过 Notification Manager 来通知用户这些事件发生了，它既可以在资源 AndroidManifest.xml 中注册，也可以在代码中通过 Context.registerReceiver() 进行注册在 AndroidManifest.xml 中注册以后，当事件来临时，即使程序没有启动，系统也会自动启动



此应用程序。另外，各应用程序可以很方便地通过 `Context.sendBroadcast()` 将自己的事件广播给其他应用程序。

### 3. Service

Android 中的 Service 和 Windows 中的 Service 完全是一个概念，用户可以通过 `StartService(Intent service)` 启动一个 Service，也可通过 `Context.bindService` 来绑定一个 Service。

### 4. Content Provider

Content Provider 提供了应用程序之间数据交换的机制，一个应用程序通过实现一个 Content Provider 的抽象接口将自己的数据暴露出去，并且隐蔽了具体的数据存储实现，这样实现了 Android 应用程序内部数据的保密性。标准的 Content Provider 提供了基本的 CRUD (Create、Read、Update、Delete) 接口，并且实现了权限机制，保护了数据交互的安全性。

一个标准的 Android 应用程序的工程文件包含以下几大部分。

- ❑ Java源代码部分（包含Activity）——放置在src目录中。
- ❑ R.java文件——由Eclipse自动生成与维护，开发者不需要修改，提供了对Android资源的全局索引。
- ❑ Android Library——应用程序运行的Android库。
- ❑ assets目录——主要用于放置多媒体等文件。
- ❑ res目录——放置的是资源文件：`drawable`包含的是图片文件，`layout`里面包含的是布局文件，`values`里面主要包含的是字符串（`strings.xml`）、颜色（`colors.xml`）以及数组（`arrays.xml`）资源。
- ❑ `AndroidManifest.xml`——应用的配置文件，在这个文件中，需要声明所有用到的 Activity、Service、Receiver等。

## 1.4.2 数据存储

Android 中提供的存储方式包括：SharedPreferences、文件存储、SQLite 数据库存储方式、内容提供者方式 Content Provider 以及网络方式 5 种，具体介绍如下。

### 1. SharedPreferences

作为 Android 提供的一种配置文件读/写方式，默认保存在应用的目录 `data/<package name>/shared_prefs` 下，通过方法 `getSharedPreferences(xx, 0)` 来获取 SharedPreferences 对象进行读/写操作。

### 2. 文件存储

Android 系统提供了进行数据读/写访问的 API，例如，`openFileInput`、`openFileOutput` 等，需要特别注意的是，Android 中应用程序的数据是私有的，也就是说，其他应用程序无法访问当前应用程序产生的文件。





### 3. SQLite 数据库存储方式

通过继承 SQLiteOpenHelper 类提供的 CRUD 接口来进行数据库操作,方便了应用程序的数据存储操作。

### 4. 内容提供者方式 (Content Provider)

通过调用其他应用程序的数据接口来实现数据的读/写访问。

### 5. 网络方式

主要通过下面要提到的网络访问该网络提供的网络服务接口,实现数据的读/写服务(如 Webservice 数据访问接口)。

## 1.4.3 网络访问方面

Android 主要通过 java.net.\*及 Android.net.\*来进行 HTTP 访问技术的封装;利用其提供的 HttpPost、DefaultHttpClient、HttpResponse 等类提供的访问接口来实现具体的 Web 服务访问。

## 1.4.4 开发流程

软件开发流程 (Software Development Process) 即软件设计思路和方法的一般过程,包括设计软件的功能及实现的算法和方法,软件的总体结构设计和模块设计、编程和调试、程序联调和测试以及编写、提交程序。软件开发生命周期如图 1-4 所示。

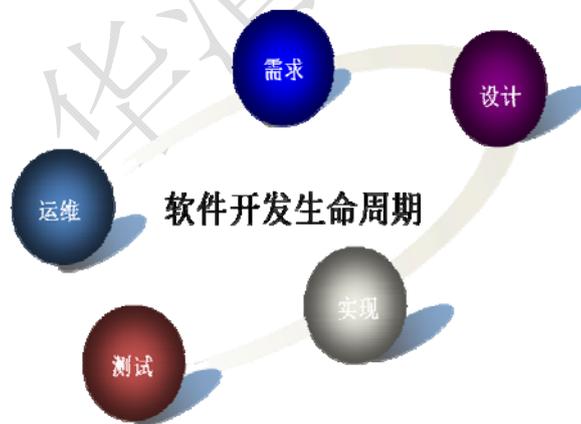


图 1-4 软件开发生命周期

需求分析:根据客户的要求,清楚了解客户需求中的产品功能、特性、性能、界面和具体规格等,然后进行分析,确定软件产品所能达到的目标。

设计:根据需求分析的结果,考虑如何在逻辑、程序上去实现所定义的产品功能、特性等,可以分为概要设计和详细设计,也可以分为数据结构设计、软件体系结构设计、应用接口设计、模块设计、界面设计等。



**编程（实现）：**开始具体的编写程序工作，分别实现各模块的功能，从而实现对目标系统的功能、性能、接口、界面等方面的要求，将设计转换成计算机可读的形式。

**测试：**对设计、编程进行验证，对用户需求进行确认。

**维护：**维持软件运行、修改软件缺陷、增强已有功能、增加新功能、升级等。

对一个软件产品或者一项软件工程来说，参与角色通常包括以下几种。

- ❑ **高级经理：**参与项目过程中各个关键环节的活动，关注产品开发的进度，对风险控制、资源提供做出决策。
- ❑ **产品经理或项目经理：**客户方和公司内部交流的纽带，主要对项目过程进行监控，对项目的进度、质量负责，制定计划、协调资源、关注和控制计划进度、控制客户期望值等。
- ❑ **开发经理：**负责界定需求，确定适当的技术构架和体系，保证软件产品按照设计的标准开发。
- ❑ **设计师：**通常设计师可以分需求分析师、构架设计师、业务设计师 3 种，能够准确把握客户需求并提供可行的实现思路，进行需求分析、进行构架设计和功能设计，按照规范编写相应的文档，将设计思路传播给开发人员、测试人员。
- ❑ **测试经理及测试人员：**测试经理主要负责计划和组织测试人员对目标产品进行测试，发现bug、跟踪bug，直到解决bug；测试人员则根据测试经理的计划和测试总体方案对目标产品进行测试，编写测试用例和测试代码，发现和跟踪bug；编写用户手册；进行用户培训和教育。
- ❑ **开发人员：**根据设计师的设计成果进行具体编码工作，对自己的代码进行基本的单元测试。
- ❑ **项目实施人员：**负责软件系统安装配置、系统割接、运行期间的维护工作。

## 1.5 OMS 简介

OMS 是 Open Mobile System 的简称，是中国移动和 Google 联合开发的一款开放式手机操作系统。该系统基于 Linux 内核，为 Android 平台而构建，并在原有的 Android 平台基础上针对中国市场进行特殊优化，内置了中国移动的服务菜单、音乐随身听、手机导航、号簿管家、139 邮箱、飞信、快讯和移动梦网等特色业务，以满足中国市场的需求。

### 1.5.1 OPhone 介绍

联想的 OPhone 手机是首款使用 OMS 系统的手机。OPhone 是基于 Linux 的面向移动互联网的终端基础软件及系统解决方案。为了突破 TD 终端“瓶颈”，以及促进手机终端与中国移动的网络及应用服务进行无缝对接，中国移动在 Google Android 操作系统的基础上，





基于 Linux 内核，推出了“深度定制”的移动操作系统 OPhone，其在业务层和用户体验层与此前的谷歌手机不完全一样。目前来讲，OPhone 与 Android 是兼容的。

下面介绍 OPhone 和 Android 的不同。

- 移动业务层面：在终端手机上完整深度定制了多种中国移动数据业务，例如，飞信、快讯、无线音乐随身听、139 邮箱、移动梦网、号簿管家、百宝箱等。使中国移动的数据业务第一次和手机的自身用户体验达到深度结合，例如，电话本中可以探测出好友飞信的在线状态，音乐播放器本地和网络的用户体验完全一致，短信中如果收到邮件地址可以用 139 邮箱直接回复，移动梦网的浏览器和普通网页的浏览器完全相同等。
- 手机基本功能：在手机基本通信功能上也结合了中国人的使用习惯并集成了很多品牌优秀的地方。例如，手写输入和拼音T9键盘的集成，拼音和手写的切换，拨号键盘可以用拼音直接调出联系人，对话模式和文件夹模式可以随意选择短信息用户界面，彩信和短信结合的信息操作逻辑，还有其可以随意定制的主屏幕，绚丽的动画及奇妙的解锁方式等都体现了OMS 对手机基本功能的重视。
- 用户体验层面：结合中国人自己的用户行为和喜好方式，并吸取了iPhone、Android、Windows Mobile、Nokia、Black Berry 等多种移动终端的优势，设计出了完全区别于Android 的用户界面。其主要特点是大屏幕全触摸的操作风格，面向移动互联网应用的设计理念。
- OMS 的开放：除了易用、美观的界面，API的开发和兼容是OMS 另一大核心竞争力。开发者可以在OMS 上开发多种平台API 的小工具，例如，OMS 可以兼容iPhone、Android、S60、Windows Mobile 等小工具的使用，OMS 是一个百家争鸣，百花齐放的平台，为移动互联网的发展推波助澜。

### 1.5.2 Widget 介绍

Widget 是指小工具，比如，Windows Vista 系统界面中的侧边栏、博客网页中的侧边栏等。而手机平台也有对应的 Widget，它是安装在手机上的小容量客户端应用程序。例如，桌面时钟、实时天气等。以下主要介绍手机平台上的 Widget。

Widget 不仅兼容多种网页技术（XHTML、CSS、JavaScript、AJAX），而且还可以通过三星 Widget API 获取各种手机功能。通常 Widget 主要用于应付简单任务，如提供网站搜寻界面。执行简单计算、显示新闻标题、天气预报、提供股票行情等。此外，Widget 还可以充当提供多重功能的复合应用程序，Widget 的开发和手机安装是十分快捷便利的。

Widget 在 Widget 系统托盘条中都有自己的托盘图标，都可以通过 Widget 包格式来实现分发，都拥有 XHTML 文件、CSS 样式表、JavaScript 代码及图像文件。Widget 可以分为在线 Widget 与离线 Widget。在线 Widget 通过手机数据连接（WiFi/3G/EDGE/GPRS）来获取内容/数据，而离线 Widget 只能使用用户生成的数据或存储在手机中或 Widget 中固定编程的数据。



## 1.6 本章小结

本章从介绍 Android 基本概念开始, 让读者了解了 Android 平台特性、Android 系统架构、Android 开发框架, 以及 OMS 等知识, 使读者可以对 Android 的应用前景有一个很好的认识。

### 关键知识点测评

1. 以下有关 Android 平台的说法, 不正确的一个是 ( )。
  - A. Android 平台具有传统的语音通话功能
  - B. Android 具有短消息功能, 以及通常移动电话都具有的个人信息系统管理方面的功能
  - C. Android 平台提供了 USB、GPS、红外、蓝牙、无线局域网等多种连接方式
  - D. Android 平台不能自定义手机的功能
2. 以下有关 Android 的叙述中, 正确的一个是 ( )。
  - A. Android 系统自上而下分为三层
  - B. Android 系统在核心库层增加了内核的驱动程序
  - C. Android 包含一个 C/C++库的集合, 以供 Android 系统的各个组件使用。这些功能通过 Android 的应用程序框架 (Application Framework) 展现给开发者
  - D. Android 的应用程序框架包括 Dalvik 虚拟机及 Java 核心库, 提供了 Java 编程语言核心库的大多数功能
3. 以下有关 Android 程序库层的叙述, 不正确的一个是 ( )。
  - A. 系统 C 库是专门为基于嵌入式 Linux 的设备定制的库
  - B. 媒体库支持播放和录制多种流行的音频和视频格式以及多种媒体格式的编码/解码格式
  - C. SGL 是 Skia 图形库, 基本的 3D 图形引擎
  - D. FreeType 包含位图 (Bitmap) 和矢量 (Vector) 字体渲染
4. 以下有关 Android 开发框架的描述, 正确的是 ( )。
  - A. 一般而言, 一个标准的 Android 程序包括 Activity、Broadcast Intent Receiver、Service、Content Provider 四部分
  - B. Android 中的 Service 跟 Windows 当中的 Service 不同
  - C. Broadcast Intent Receiver 提供了应用程序之间数据交换的机制
  - D. Content Provider 为不同的 Activity 进行跳转提供了机制
5. 下列有关 OMS 的描述, 正确的一个是 ( )。
  - A. OPhone 与 Android 的用户界面相同
  - B. OMS 是在原有的 Android 平台基础上针对中国市场进行特殊优化, 以满足中国市场的需要
  - C. OMS 不能兼容 iPhone、Android、S60、Windows Mobile 等小工具的使用



D. OPhone 在业务层和用户体验层与此前的谷歌手机相同

## 资源分享:

华清远见 Android 开发教程下载:

<http://download.farsight.com.cn/download/android.htm>

华清远见 2012 更多 Android 相关新版教材:

<http://www.3g-edu.org/courses/book.htm>

## 相关课程推荐:

Android 系统开发入门实战体验日 (1 天):

<http://www.farsight.com.cn/courses/TS-3GBasic.aspx>

3G Android 系统开发就业培训班 (4 个月):

<http://www.3g-edu.org/3g/>

Android 应用开发培训班 (短期班):

<http://www.farsight.com.cn/courses/TS-AndroidApp.htm>

Android 系统开发培训班 (短期班):

<http://www.farsight.com.cn/courses/TS-Android.htm>

## 联系我们:

免费电话: 800-810-3930, 400-706-1880

华清远见集团官网: [www.hqvj.com](http://www.hqvj.com)

华清远见企业学院: [www.farsight.com.cn](http://www.farsight.com.cn)

华清远见嵌入式学院: [www.embedu.org](http://www.embedu.org)

华清远见 3G 学院: [www.3g-edu.org](http://www.3g-edu.org)

华清远见物联网学院: [www.topsight.cn](http://www.topsight.cn)

华清远见研发中心: [dev.hqvj.com](http://dev.hqvj.com)



华清远见 3G 学院: [www.3g-edu.org](http://www.3g-edu.org)

