



10年口碑积累，成功培养50000多名研发工程师，铸就专业品牌形象

华清远见的企业理念是不仅要良心教育、做专业教育，更要做受人尊敬的职业教育。

# 《Android 系统下 Java 编程详解》

作者：华清远见

专业始于专注 卓识源于远见

## 第 1 章 Android 基本概念

---

本章简介

---

本章主要介绍 Android 基本概念方面的内容，包括 Android 平台特性、Android 系统架构、Android 开发框架和 Android 开发环境搭建。

## 1.1 Android 简介

Android 一词英文本义指“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 公司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，逐渐发展成为当下流行的手机开发平台，图 1-1 所示为 Android 系统的 Logo。



图 1-1 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程序的移动设备软件堆。换言之，Android 是基于 Java 并运行在 Linux 内核上的轻量级操作系统，其功能全面，包括一系列 Google 公司在其上内置的应用程序，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 的模拟器



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）正朝着智能化的方向发展，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为通信功能，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，移动电话通常被作为网络工具使用，即移动电话可以完成电脑的部分功能。而 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。

### 3. 媒体播放器

随着多媒体技术的发展应用，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，可更流畅地播放视频，以及支持网络流媒体等功能。

### 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 平台针对于用户的个性化需求，提供了全面自定义手机的功能。

除了上述介绍的 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 4 层：Linux 内核、Android 程序库及 Android 运行时、Android 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 内核之外，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、Wi-Fi 驱动、音频系统驱动、电源管理等，这些驱动程序为 Android 系统的运行提供基础性支持。

这样分层的好处是在使用下层提供的服务的同时也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 程序库 (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常见的音频和视频格式，以及多种媒体的编码/解码格式，包括 MPEG-4、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 1.0 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 (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 应用程序都是 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中可以高效地运行多个虚拟机。Dalvik 虚拟机可执行文件格式是 .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指令，而基于寄存器的机器指令更大。dex 是一套工具，可以将 Java 的 .class 格式转换成 .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，包括如下内容。

- ❑ 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 (Lists)、网格 (grids)、文本框 (Textboxes)、按钮 (Buttons)，甚至是可嵌入的网页浏览器。
- ❑ 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。
- ❑ 消息管理器 (Notification Manager)——使得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 应用程序和小部件

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，其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。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 3 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。对于移植开发移动电话系统主要由移动电话的制造者来进行，其产品主要是 Android 手机；而公司、团体和个人一般进行对于 Android 应用程序的开发，产生各种各样的 Android 应用程序。

对于 Android 移植开发来讲，其主要工作集中于 Linux 内核中的相关设备驱动程序，以及 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 4 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 类中主要是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、事件的响应设计及数据处理设计、导航设计等 application 设计的方方面面。

因此，Activity 是手机上的一个界面，相当于一个网页，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 间进行跳转提供了机制，比如当从 A Activity 跳转到 B Activity，使用 Intent 来实现如下：

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

BroadcastReceiver 还为各种不同的 Android 应用程序间进行进程间的通信提供了可能。例如，当电话呼入时，它可以通过 BroadcastReceiver 发布广播消息。对于用户而言，用户是无法看到 BroadcastReceiver 事件的，它对用户是不透明的，BroadcastReceiver 通过 NotificationManager 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，注册以后，当事件来临时，即使程序没有启动，系统也会在需要的时候自动启动此应用程序；另外，各应用程序很方便地通过 Context.sendBroadcast() 将自己的事情广播给其他应用程序。

#### 3. Service

Android 中的 Service 和 Windows 中的 Service 是一个概念，用户可以通过 startService (Intent Service) 启动一个 Service，也可以通过 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) 的接口，并且实现了权限机制，保障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 目录当中。
- ❑ R.java 文件——由 Eclipse 自动生成与维护的，开发者不需要修改，提供了 Android 对的资源全局索引。
- ❑ Android Library——应用程序运行的 Android 库。
- ❑ assets 目录——主要用于放置多媒体等一些文件。
- ❑ res 目录——放置的是资源文件：drawable 包含图片文件、layout 包含布局文件，values 目录里面主要包含字符串 (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

在 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) 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,特别需要注意的是,在 Android 中应用程序的数据是私有的,也就是说,当前应用程序产生的文件其他应用程序无法访问。

## 3. SQLite 数据库存储方式

通过继承 `SQLiteOpenHelper` 类,通过该类提供的 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

主要是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(例如,WebService 数据访问接口)。

### 1.4.3 网络访问方面

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

## 1.5 Android 开发环境搭建

Android 应用软件开发需要的开发环境如表 1-1 所示。

表 1-1 Android 应用软件开发需要的开发环境

所需项	版本需求	说明	备注
操作系统	Windows XP 或 Vista Mac OS X10.4.8+ Linux Ubuntu Drapper		选择自己最熟悉的操作系统
软件开发包	Android SDK	选择最新版本的 SDK	截止到目前,其最新版本为 2.3
IDE	Eclipse IDE+ADT	Eclipse3.3 (Europa), 3.4 (Ganymede) ADT (Android Development Tools) 开发插件	选择 “for Java Developer”
其他	JDKApache Ant	Java SE Development Kit 5 或 6 Linux 和 Mac 上使用 Apache Ant 1.6.5+, Windows 上使用 1.7+版本	JDK 和 SRE 一般同时安装;不兼容 Gnu Java 编译器 (gcj)

软件开发包的下载地址如下:

- (1) JDK 1.6, <http://java.sun.com/javase/downloads/index.jsp>。
- (2) Eclipse 3.4 (Eurpa), <http://www.eclipse.org/downloads/下载 Eclipse IDE for Java Developers>。
- (3) Android SDK 2.2, <http://developer.android.com>。

### 1.5.1 安装 JDK 和配置 Java 开发环境

首先下载使用 JDK 的包，并进行安装。例如，得到 JDK 1.6 版本的安装文件 jdk-6u10-rc2-bin-b32-windows-i586-p-12\_sep\_2008.exe，双击进行安装。接受许可证，选择需要安装的组件和安装路径后，单击“下一步”按钮，完成安装过程。

安装完成后，利用以下步骤检查安装是否成功：打开 CMD 窗口，在 CMD 窗口中输入 `java -version` 命令，如果屏幕出现如图 1-4 所示的代码信息，说明 JDK 安装成功。

```
java version "1.6.0_26"
Java(TM) SE Runtime Environment (build 1.6.0_26-b03)
Java HotSpot(TM) Client VM (build 20.1-b02, mixed mode, sharing)
```

图 1-4 JDK 安装检查

### 1.5.2 Eclipse 的安装

JDK 安装成功后，可以直接安装 Eclipse，例如，使用 Eclipse 3.6，得到其压缩包 eclipse-SDK-3.6.1-win32.zip，该包不需要安装，直接解压即可执行其中的 eclipse.exe 文件进行使用，Eclipse 可以自动找到用户前期安装的 JDK 路径。

### 1.5.3 SDK 和 ADT 的安装和配置

JDK 和 Eclipse 安装成功后，下载安装 Android 的 SDK，得到 android-sdk-windows.zip，解压后运行 SDK Manager.exe，选择需要的 API 版本进行安装，如图 1-5 所示。

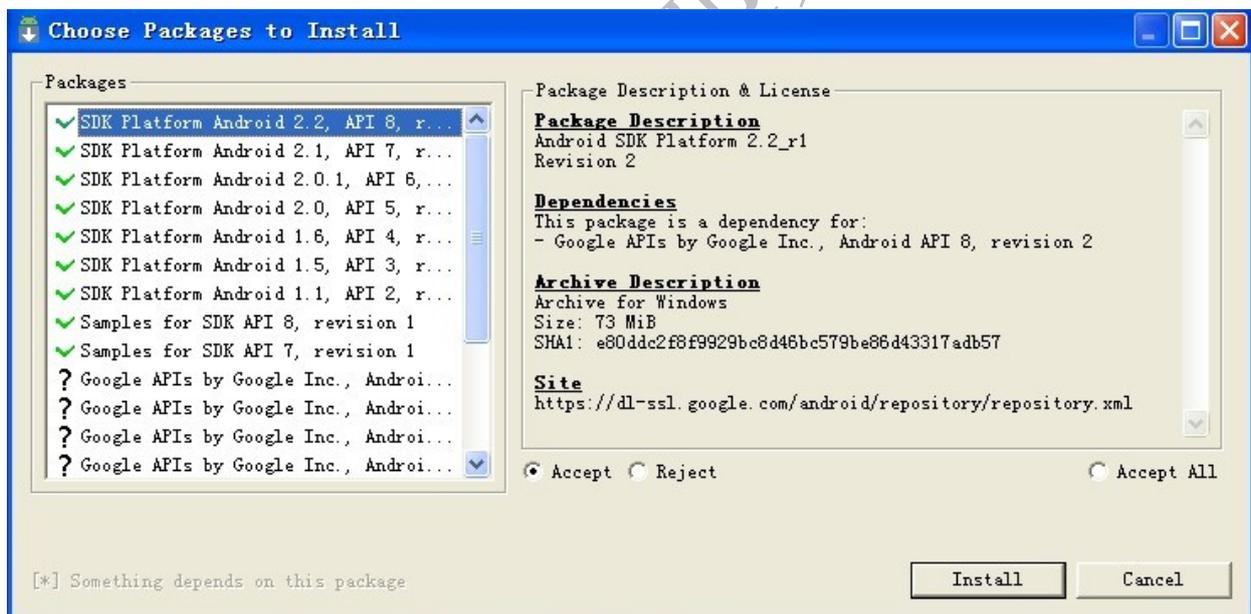


图 1-5 SDK Manager

完成以上步骤后，在 Eclipse 中安装配置 ADT。

ADT (Android Development Tools) 是 Android 为 Eclipse 定制的一个插件，为用户提供了一个强大的用户开发 Android 应用程序的综合环境。ADT 扩展了 Eclipse 的功能，可以让用户快速地建立 Android 项目，创建应用程序界面，在基于 Android 框架 API 的基础上添加组件，以及用 SDK 工具集调试应用程序，甚至导出签名（或未签名）的 APKs 以便发行应用程序。

在 Eclipse 中安装 ADT，首先启动 Eclipse，选择“Help”→“Software Updates”命令，准备安装插件。在打开的 InStall 窗口中选择 Available Software，得到如图 1-6 所示的界面。

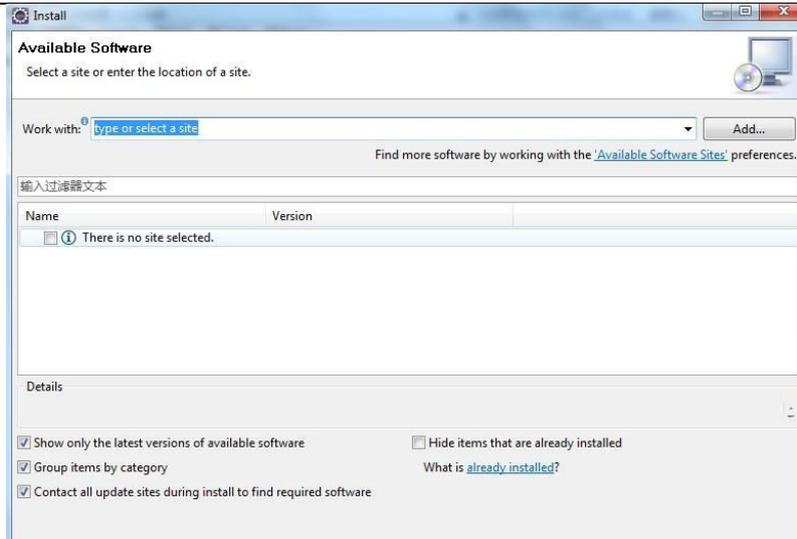


图 1-6 Available Software 界面

单击 Add 按钮，得到如图 1-7 所示的界面。



图 1-7 Add Site 界面

在 Location 文本框中，输入 Android 插件的路径：`http://dl-ssl.google.com/android/eclipse/`，单击“OK”按钮，返回至图 1-6 所示界面，可以看到刚刚添加的站点已添加至搜索列表，选择到 Developer Tools 中，然后单击“Install”按钮。在 Install 界面中，选择 Android DDMS 和 Android Development Tools，单击“Next”按钮，阅读并接受许可协议，单击 Finish 按钮完成安装。安装完成后重启 Eclipse 即可。

在 Eclipse 中增加 SDK 的路径：选择“Windows”→“Preference”命令，在左侧的列表中选择 Android 选项，设置 SDK Location 为 SDK 的目录，如图 1-8 所示。

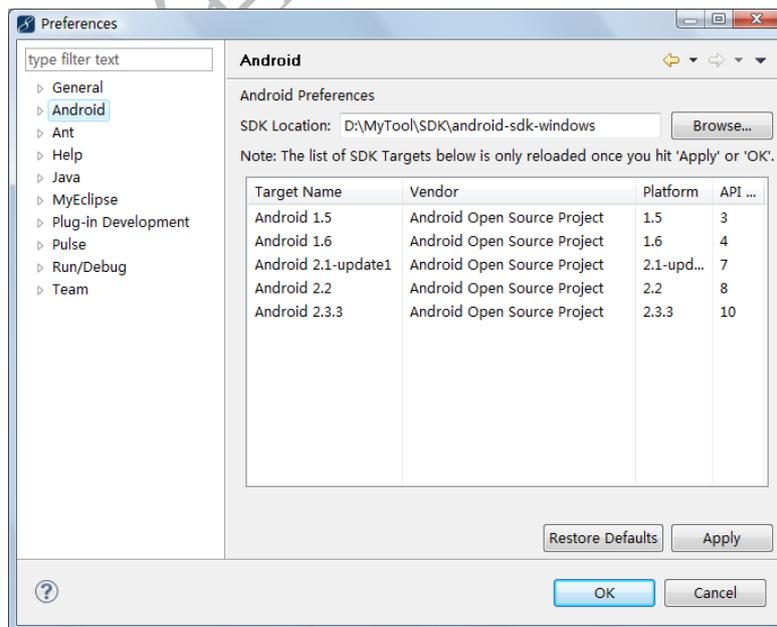


图 1-8 选择 Android SDK 路径

问：安装 ADT 的过程中没有网络怎么办？

答：在“Add Site”对话框中单击 Archive 按钮，直接指定磁盘中的 ADT 包（例如，ADT-0.9.1.zip 等）。

## 1.6 本章小结

本章主要介绍 Android 的基本概念，包括平台特性、系统架构、开发框架等；介绍了 Android 开发环境的搭建。读者通过本章的学习，对 Android 开发有了初步认识，在搭建好 Android 开发环境的同时为后续章节的学习做好准备。

### 课后练习题

#### 一、选择题

- 以下有关 Android 平台的说法，不正确的是（ ）。
  - Android 平台具有传统的语音通话功能
  - Android 具有短消息功能及通常移动电话都具有的个人信息系统管理方面的功能
  - Android 平台提供了 USB、GPS、红外、蓝牙、无线局域网等多种连接方式
  - Android 平台不能自定义手机的功能
- 以下有关 Android 的叙述，正确的是（ ）。
  - Android 系统自上而下分为三层
  - Android 系统在核心库层增加了内核的驱动程序
  - 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）暴露给开发者
  - 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
- 以下有关 Android 程序库层的叙述，不正确的是（ ）。
  - 系统 C 库是专门为基于嵌入式 Linux 的设备定制的库
  - 媒体库支持播放和录制多种常见的音频和视频格式及多种媒体格式的编码/解码格式
  - SGL 是 Skia 图形库，基本的 3D 图形引擎
  - FreeType 包含位图（bitmap）和矢量（vector）字体渲染
- 以下有关 Android 开发框架的描述，正确的是（ ）。
  - 一般而言，一个标准的 Android 程序包括 Activity、Broadcast Intent Receiver、Service、Content Provider 四部分
  - Android 中的 Service 和 Windows 中的 Service 不同
  - Broadcast Intent Receiver 提供了应用程序之间数据交换的机制
  - Content Provider 为不同的 Activity 间进行跳转提供了机制
- 以下有关 Android 开发环境所需条件的说法，不正确的是（ ）。
  - 可在 Windows/Linux 操作系统上进行开发
  - 使用 Eclipse IDE 进行开发
  - 需在 Eclipse IDE 中安装配置 ADT
  - 可以只安装 JRE

#### 二、简答题

简述 Android 应用程序框架的层次结构。

## 联系方式

集团官网：[www.hqyj.com](http://www.hqyj.com)

嵌入式学院：[www.embedu.org](http://www.embedu.org)

移动互联网学院：[www.3g-edu.org](http://www.3g-edu.org)

企业学院: [www.farsight.com.cn](http://www.farsight.com.cn)

物联网学院: [www.topsight.cn](http://www.topsight.cn)

研发中心: [dev.hqyj.com](http://dev.hqyj.com)

集团总部地址: 北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址: 北京市海淀区西三旗悦秀路北京明园大学校区, 电话: 010-82600386/5

上海地址: 上海市徐汇区漕溪路 250 号银海大厦 11 层 B 区, 电话: 021-54485127

深圳地址: 深圳市龙华新区人民北路美丽 AAA 大厦 15 层, 电话: 0755-22193762

成都地址: 成都市武侯区科华北路 99 号科华大厦 6 层, 电话: 028-85405115

南京地址: 南京市白下区汉中路 185 号鸿运大厦 10 层, 电话: 025-86551900

武汉地址: 武汉市工程大学卓刀泉校区科技孵化器大楼 8 层, 电话: 027-87804688

西安地址: 西安市高新区高新一路 12 号创业大厦 D3 楼 5 层, 电话: 029-68785218

华清远见