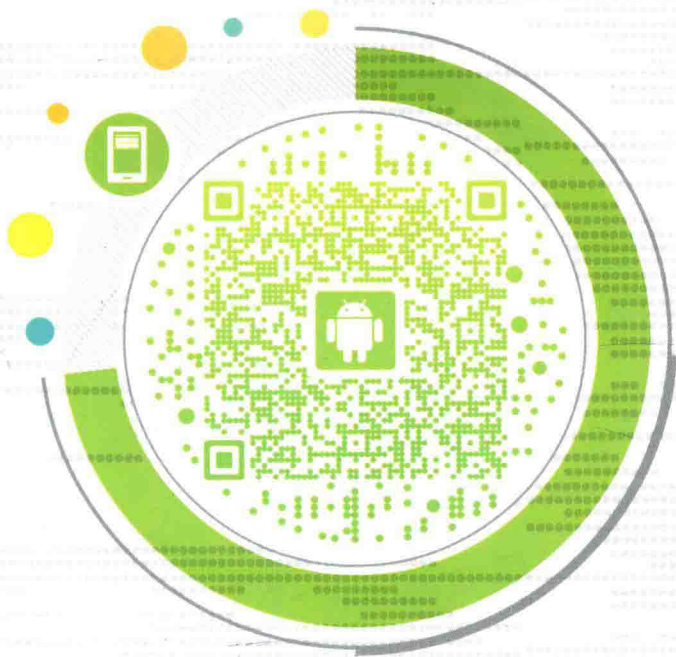


21世纪高等学校计算机类课程创新规划教材 · 微课版



Android Studio 应用程序设计 (第2版)

◎ 张思民 编著

微课版

5 HOURS

5小时
教学视频

零基础入门

项目式教学

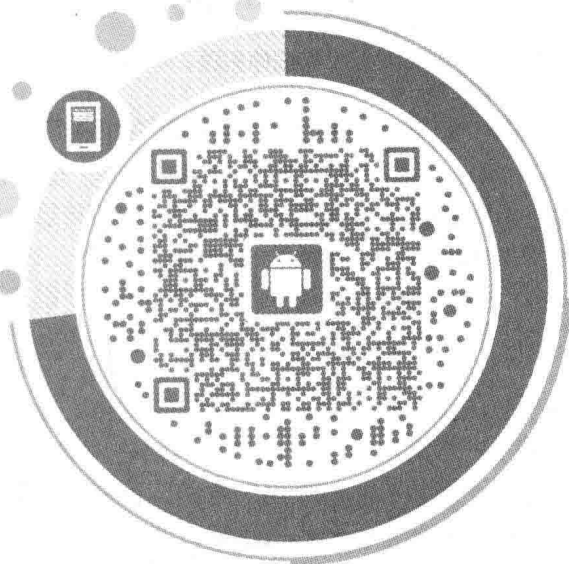
源码和课件

案例演示视频

清华大学出版社



21世纪高等学校计算机类课程创新规划教材 · 微课版



Android Studio 应用程序设计 (第2版)

◎ 张思民 编著

微课版

清华大学出版社
北京

内 容 简 介

本书是面向 Android Studio 初学者的入门教程，内容基本涵盖了 Android 相关的所有技术，大致可以分为两个部分。第一部分（第 1~4 章）主要介绍 Android Studio 的安装、应用程序的结构、图形用户界面的组件及其设计方法；第二部分（第 5~9 章）主要介绍较高级的主题，内容包括后台服务与系统服务、网络通信技术、数据存储技术、地图服务与传感器检测技术等。

本书内容浅显易懂，对每一个知识点都配了相应的例题。本书提供了电子课件和所有例题的源代码，扫描每章提供的二维码可观看教学视频。

本书可作为高等院校及各类培训学校 Android 系统课程的教材，也可作为希望学习 Android 系统开发的读者的自学用书。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目（CIP）数据

Android Studio 应用程序设计：微课版/张思民编著. —2 版. —北京：清华大学出版社，2017
(21 世纪高等学校计算机类课程创新规划教材·微课版)

ISBN 978-7-302-48134-8

I. ①A… II. ①张… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字（2017）第 203703 号

责任编辑：魏江江 王冰飞

封面设计：刘 键

责任校对：胡伟民

责任印制：李红英

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社总机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载：<http://www.tup.com.cn>, 010-62795954

印 装 者：清华大学印刷厂

经 销：全国新华书店

开 本：185mm×260mm 印 张：17.75 字 数：435 千字

版 次：2013 年 3 月第 1 版 2017 年 10 月第 2 版 印 次：2017 年 10 月第 1 次印刷

印 数：9501~11500

定 价：39.50 元

产品编号：075773-01

第2版前言

与第1版相比，第2版的最大修改之处是把开发工具 Eclipse 换成了 Android Studio。这主要考虑 Android Studio 是 Google 公司推出的专门为 Android “量身定制”的开发工具，是 Google 大力支持的一款基于 IntelliJ IDEA 改造的 IDE 集成开发环境，可以说它是 Android 开发工具的未来。

第2版删除了第1版中 Java 语言基础知识简介的章节，增加了访问 Web 服务器数据的内容。另外，由于 Android Studio 与 Eclipse 的目录结构和菜单差异很大，因此很多例题在介绍操作时做了修改。

第2版全书共分9章，第1章主要讲解 Android Studio 开发环境的搭建，并介绍了开发 Android 应用程序的步骤和应用程序框架的结构；第2~3章讲解如何使用布局和视图创建用户界面，介绍了用户图形界面的常用组件；第4章介绍图形与多媒体处理技术，介绍了绘制几何图形的基本方法、处理触摸屏事件的方法，还详细讨论了音频播放和视频播放的设计以及录音、照相技术，并详细讲解了在 Android 中实现动画的技术；第5章介绍后台服务与系统服务，以及系统功能调用；第6章介绍网络通信技术，介绍了 Web 视图以及基于 TCP 协议的网络程序设计、基于 HTTP 协议的网络程序设计等网络编程技术；第7章介绍应用 Volley 框架访问 Web 服务器，并介绍了 JSON 数据格式和一个网络音乐播放器设计实例；第8章介绍数据存储技术，介绍了 SQLite 数据库存储方式、文件存储方式和 XML 文件的 SharedPreferences 存储方式，还介绍了访问远程数据库的方法；第9章讲解地图服务与传感器检测技术，地图服务主要介绍实现地图视图的基本方法，传感器检测主要介绍重力加速度的应用。

本书提供了电子课件和所有例题的源代码，扫描每章提供的二维码可观看教学视频。

张思民

2017年5月

目 录

第 1 章 Android 系统及其开发过程	1
1.1 Android 系统概述	1
1.1.1 Android 系统及特点	1
1.1.2 Android 系统的体系结构	2
1.1.3 Android 开发的分类	3
1.2 搭建 Android Studio 开发环境	4
1.2.1 安装 Android Studio 前的必要准备	4
1.2.2 安装 Android Studio 详解	5
1.2.3 设置环境变量	7
1.3 Android API 和在线帮助文档	8
1.4 Android 应用程序的开发过程	9
1.4.1 开发 Android 应用程序的一般过程	9
1.4.2 生成 Android 应用程序框架	10
1.4.3 编写代码生成 MainActivity.java	12
1.4.4 在模拟器中运行应用程序	13
1.5 Android 项目结构	13
1.5.1 目录结构	13
1.5.2 Android 应用程序结构分析	19
1.6 Android 应用程序设计示例	21
习题 1	23
第 2 章 Android 用户界面的设计	24
2.1 用户界面组件包 widget 和 View 类	24
2.2 文本标签 TextView 与按钮 Button	25
2.2.1 文本标签	25
2.2.2 按钮及按钮处理事件	27
2.3 文本编辑框	32
2.4 Android 布局管理	35
2.4.1 布局文件的规范与重要属性	36
2.4.2 常见的布局方式	37
2.5 进度条和选项按钮	46

2.5.1	进度条	46
2.5.2	选项按钮	48
2.6	图像显示类 ImageView 与画廊组件类 Gallery	55
2.6.1	图像显示类 ImageView	55
2.6.2	画廊组件类 Gallery 与图片切换器 ImageSwitcher	59
2.7	消息提示类 Toast	64
2.8	列表组件	67
2.8.1	列表组件类 ListView	67
2.8.2	ListActivity 类	69
2.9	滑动抽屉组件类 SlidingDraw	72
	习题 2	76
第 3 章	多个用户界面的程序设计	78
3.1	页面的切换与传递参数值	78
3.1.1	传递参数组件 Intent	78
3.1.2	Activity 页面的切换	78
3.1.3	应用 Intent 在 Activity 页面之间传递数据	82
3.2	菜单设计	86
3.2.1	选项菜单	86
3.2.2	上下文菜单	89
3.3	对话框	91
3.3.1	消息对话框 AlertDialog	91
3.3.2	其他几种常用对话框	96
	习题 3	100
第 4 章	图形与多媒体处理	101
4.1	绘制几何图形	101
4.1.1	几何图形绘制类	101
4.1.2	几何图形的绘制过程	102
4.1.3	自定义组件	107
4.2	触摸屏事件的处理	110
4.2.1	简单的触摸屏事件	110
4.2.2	手势识别	116
4.3	音频播放	120
4.3.1	多媒体处理包	120
4.3.2	多媒体处理播放器 MediaPlayer	120
4.3.3	播放音频文件	121
4.4	视频播放	127
4.4.1	应用媒体播放器播放视频	127

4.4.2	应用视频视图播放视频	130
4.5	录音与拍照	133
4.5.1	用于录音、录像的 MediaRecorder 类	133
4.5.2	录音示例	134
4.5.3	拍照	137
4.6	动画技术	142
4.6.1	动画组件类	142
4.6.2	补间动画 Tween Animation	143
4.6.3	属性动画 Property Animation	148
习题 4		152
第 5 章	后台服务与系统服务	153
5.1	后台服务 Service	153
5.2	信息广播机制 Broadcast	157
5.3	系统服务	166
5.3.1	Android 的系统服务	166
5.3.2	系统通知服务 Notification	167
5.3.3	系统定时服务 AlarmManager	169
5.3.4	系统功能的调用	172
习题 5		175
第 6 章	网络通信技术	176
6.1	Web 视图	176
6.1.1	浏览器引擎 WebKit	176
6.1.2	Web 视图对象	176
6.1.3	调用 JavaScript	180
6.2	基于 TCP 协议的网络程序设计	186
6.2.1	网络编程的基础知识	187
6.2.2	利用 Socket 设计客户机/服务器系统程序	191
6.2.3	应用 Callable 接口实现多线程 Socket 编程	196
6.3	基于 HTTP 协议网络程序设计	200
6.3.1	建立 PHP 服务器网站	200
6.3.2	应用 HttpURLConnection 访问 Web 服务器	200
习题 6		210
第 7 章	应用 Volley 框架访问 Web 服务器	211
7.1	Volley 框架及其应用	211
7.1.1	Volley 包的下载与安装	211
7.1.2	JSON 数据格式简介	212

7.1.3	Volley 的工作原理和几个重要对象	216
7.1.4	Volley 的基本使用方法	217
7.2	应用 Volley 框架设计网络音乐播放器	221
	习题 7	226
第 8 章	数据存储技术	227
8.1	SQLite 数据库	227
8.1.1	SQLite 数据库简介	227
8.1.2	管理和操作 SQLite 数据库的对象	228
8.1.3	SQLite 数据库的操作命令	228
8.2	文件的处理	240
8.2.1	输入/输出流	240
8.2.2	处理文件流	241
8.3	轻量级存储 SharedPreferences	248
8.4	访问远程数据库	250
	习题 8	255
第 9 章	地图服务与传感器检测技术	256
9.1	电子地图服务的应用程序开发	256
9.1.1	Android 地图的 SDK 开发包的下载以及 Key 的申请	256
9.1.2	显示地图的应用程序示例	259
9.2	传感器检测技术	262
9.2.1	传感器简介	262
9.2.2	加速度传感器的应用示例	264
	习题 9	273

1.1 Android 系统概述

1.1.1 Android 系统及其特点

2007 年 11 月 5 日, Google 公司推出了基于 Linux 操作系统的智能手机平台 Android 系统。Android 系统由操作系统、中间件、用户界面程序和应用软件等组成。2013 年 5 月 16 日, Google 公司推出新的 Android 开发环境——Android Studio。Android 的出现绝非偶然, 是由传统的移动电话系统开发模式演变而来的一种符合时代潮流的新型移动开发模式的产物。

Android 传奇的创造与被称为 Android 之父的 Android 创始人安迪·鲁宾 (Andy Rubin) 密不可分。2003 年, 安迪·鲁宾成立了一家叫 Android 的公司, 致力于开发一个面向所有软件设计者开放的移动手机平台。安迪·鲁宾的 Android 项目因顺利开展受到了一些风险投资公司的青睐。2005 年 3 月, 由安迪·鲁宾继续负责 Android 项目的研发工作。2007 年 11 月, Google 公司正式公布了 Android 操作系统, 并且宣布与 34 家手机厂商、运营商成立“开放手机联盟 (OHA)”, 自此这个基于 Linux 内核的 Android 系统正式登上历史舞台。在接下来的 5 年里, 安迪·鲁宾负责的 Android 系统获得了令人难以置信的成功。到 2013 年, 在市场占有率方面, Google 公司的 Android 系统主导了手机世界。

Android 系统诞生在开放时代的背景下, 其全开放的智能移动平台、多硬件平台的支持、使用众多标准化的技术、完整的核心技术、完善的 SDK 和文档、完善的辅助开发工具等特点与智能手机的发展方向紧密相连, 它将代表并引领新时代的技术潮流。

Android 系统具有开放性、平等性、方便性及硬件丰富性等特点。下面对这些特点进行简单介绍。

1. 系统开放性

Android 系统是一款真正开放的系统。Android 系统从底层的操作系统直到最上层的应用程序都是开放的, 程序开发人员和爱好者都可以很方便地从网络上获取到源代码, 可以对它们进行分析和移植。

2. 应用程序平等性

在 Android 系统开发平台上 Android 系统自带的程序与程序开发人员自己开发的应用程序都是平等的, 程序开发人员可以开发个人喜爱的应用程序来代替系统的程序, 构建个性化的 Android 系统。

3. 开发方便性

在 Android 系统开发平台上开发应用程序是非常方便的, Android 系统为开发人员提供了大量的实用组件库和方便的工具, 开发人员只需编写几行代码就可以将功能强大的组件添加到自己的程序中。

4. 硬件丰富性

由于 Android 系统的开放性, 众多的硬件制造商纷纷开发出各种各样的可以与 Android 系统兼容的产品, 进一步丰富了 Android 系统的应用。

1.1.2 Android 系统的体系结构

Android 系统的体系结构和其操作系统一样, 采用了分层的架构。Android 系统分为 4 层, 从顶层到底层分别是应用程序层、应用程序框架层、系统运行库层和 Linux 核心层, 如图 1.1 所示。

1. 应用程序

Android 系统自带了一套核心应用程序, 包括电话拨号程序、短信程序、日历、音乐播放器、浏览器、联系人管理程序等, 如图 1.2 所示。所有的应用程序都是用 Java 语言编写的, 开发人员自己开发的应用程序就位于应用程序层。该层的程序是完全平等的, 开发人员可以任意用自己开发的应用程序替换 Android 系统自带的程序。

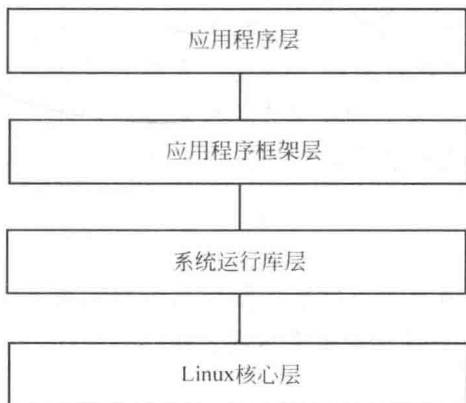


图 1.1 Android 系统的体系结构



图 1.2 Android 系统自带的应用程序

2. 应用程序框架

Android 系统通过应用程序框架为开发人员创建自己的应用程序提供了一个开放的开

发平台，程序开发人员可以在这个应用程序框架平台上设计自己的应用程序。本书所讲的设计都是基于这个应用程序框架完成设计的。

Android 系统的应用程序框架主要包含以下 9 个部分。

- 活动页面管理 (Activity Manager): 用于管理程序的生命周期。
- 窗口管理 (Window Manager): 用于管理应用程序窗口。
- 内容供应 (Content Providers): 提供数据共享, 使一个应用程序可以访问另一个应用程序的数据。
- 视图系统 (View System): 用于构建应用程序的可视化组件。
- 包管理 (Package Manager): 用于管理项目程序。
- 电话管理 (Telephone Manager): 移动设备的基本功能统一由电话管理器管理。
- 资源管理 (Resource Manager): 为应用程序提供所需的文字、声音、图片、视频或布局文件等资源。
- 位置管理 (Location Manager): 用于提供位置服务。
- 通知管理 (Notification Manager): 在手机顶部的状态栏中发布消息提示。

3. 系统运行库

1) 程序库

Android 包含一些 C/C++ 程序库, 这些库能被 Android 系统中不同的组件使用。它们通过 Android 应用程序框架为开发者提供服务。

2) Android 运行时库

Android 包含一个核心库, 该核心库提供了 Java 编程语言核心库的大多数功能。Android 系统的 Dalvik 虚拟机也包含在 Android 运行时库中。

4. Linux 内核

Android 的核心系统服务依赖于 Linux 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。

1.1.3 Android 开发的分类

对于开发者而言, Android 开发分为以下两大类。

1. 系统移植开发

移植开发是为了使 Android 系统能在手持式移动设备上运行, 在具体的硬件系统上构建 Android 软件系统。这种类型的开发在 Android 底层进行, 需要移植开发 Linux 中相关的设备驱动程序及 Android 本地框架中的硬件抽象层, 也就是需要将设备驱动与 Android 系统联系起来。Android 系统对硬件抽象层都有标准的接口定义, 在移植时只需实现这些接口即可。

2. Android 应用程序开发

应用程序开发可以基于硬件设备(用于测试的实体手机), 也可以基于 Android 模拟器。应用程序开发处于 Android 系统的顶层, 使用 Android 系统提供的 Java 框架 (API) 进行开发设计工作, 是大多数开发者从事的开发工作。本书所介绍的 Android 应用程序设计都是在这个层上进行的。

1.2 搭建 Android Studio 开发环境

1.2.1 安装 Android Studio 前的必要准备

1. Android 系统开发的操作平台

Android 系统开发的软件环境目前有两种,一种是 Eclipse + ADT(Android Development Tools 插件)系统,另一种是 Android Studio 系统。在这里主要介绍 Android Studio 系统。

Android Studio 是一个全新的基于 IntelliJ IDEA 的 Android 开发环境(IntelliJ IDEA 是一种用 Java 语言开发的集成开发环境,是被业界公认为最好的 Java 开发工具),Android Studio 提供了集成的 Android 开发工具用于应用程序的开发和调试。

在安装 Android Studio 之前需要安装 Java JDK 的环境。

2. 下载最新版本的 Android Studio 软件

读者可以到 Android Studio 官方网站“<http://developer.android.com/sdk/index.html>”免费下载最新的系统软件,如图 1.3 所示。



图 1.3 Android Studio 官方下载页面

进入下载页面以后下载对应操作系统所支持的版本,见表 1-1(以 Android Studio 2.1 版本为例)。

表 1-1 下载 Android Studio 系统安装包

安装平台	系统安装包	Size
Windows	android-studio-bundle-143.2739321-windows.exe 包含 Android SDK (推荐)	1166 MB
	android-studio-bundle-143.2739321-windows.exe	264 MB
Mac OS X	android-studio-ide-143.2739321-mac.dmg	279 MB
Linux	android-studio-ide-143.2739321-linux.zip	278 MB

1.2.2 安装 Android Studio 详解

1. 按安装向导完成 Android Studio 系统的安装

运行安装文件 android-studio-bundle-143.2739321-windows.exe, 按照安装向导完成系统的安装, 如图 1.4 所示。



图 1.4 Android Studio 系统的安装

2. 设置 Android SDK 的存放位置

安装完成后, 第一次运行 Android Studio 系统需要设置 Android SDK, 找到 SDK 的存放位置, 如图 1.5 所示。Android SDK 的存放位置也可以通过 Android Studio 应用程序的 Settings 命令设置。

3. 创建 Android 虚拟设备 AVD

Android 应用程序可以在实体手机上执行, 也可以创建一个 Android 虚拟设备 AVD (Android Virtual Device) 来测试。每一个 Android 虚拟设备 AVD 模拟一套虚拟环境来运行 Android 操作系统平台, 这个平台有自己的内核、系统图像、外观显示、用户数据区和仿真的 SD 卡等。

下面介绍如何创建一个 Android 虚拟设备 AVD。

Android Studio 集成开发环境提供了 Android Virtual Device Manager 功能, 用户可以用它来创建和调用 Android 虚拟设备 AVD。

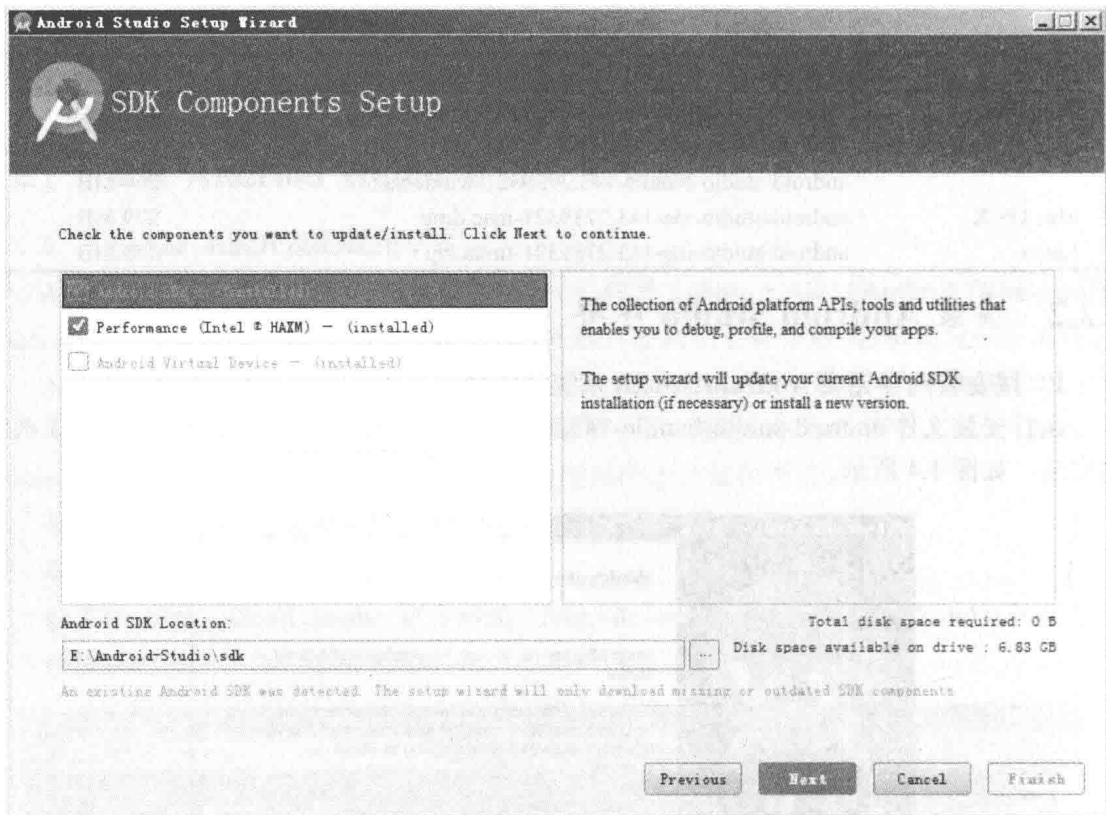


图 1.5 设置 Android SDK 的位置

(1) 选择 Android Studio 菜单栏中的 Tools→Android→AVD Manager 命令，用户在弹出的 Android Virtual Device Manager 对话框中可以看到已创建的 AVD。单击下方的 Create Virtual Device 按钮创建一个新的 AVD，如图 1.6 所示。

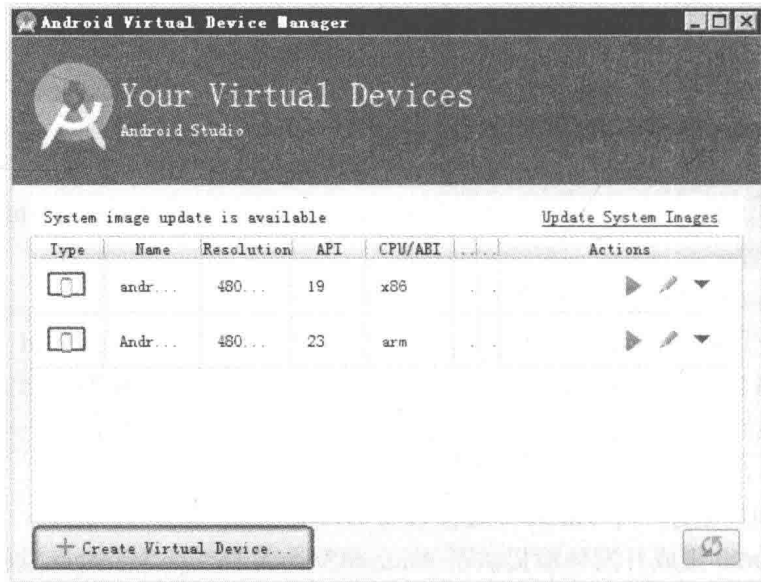


图 1.6 Android Virtual Device Manager 对话框

(2) 运行 AVD 模拟器。在 Android Virtual Device Manager 对话框中选择已经建立的 AVD，单击 Actions 栏中的 ▶ 按钮可以启动 AVD 模拟器。启动 AVD 模拟器的时间很长，建议打开后不要关闭，可以在该模拟器上测试 Android 应用程序。启动的 AVD 模拟器如图 1.7 所示。



图 1.7 Android 的 AVD 模拟器

1.2.3 设置环境变量

安装完 Android Studio 系统后还要设置环境变量，即把 Android Studio 系统目录下的 platform-tools 路径设置到系统变量中。右击桌面上的“计算机”图标，在弹出的快捷菜单中选择“属性”命令，在“控制面板\系统”对话框中选择“高级系统设置”选项，再单击“环境变量”按钮，在打开的“环境变量”对话框的“系统变量”下方找到 Path 变量，单击“编辑”按钮，在“编辑系统变量”对话框的“变量值”栏中输入 Android Studio 安装目录下的 platform-tools 完整路径和 tools 完成路径，如图 1.8 所示。

例如设安装路径如下：

```
C:\Users\Administrator\AppData\Local\Android\sdk
```

则需要增加设置 Path 变量的值：

```
C:\Users\Administrator\AppData\Local\Android\sdk\tools
```

以及

C:\Users\Administrator\AppData\Local\Android\sdk\ platform-tools

8

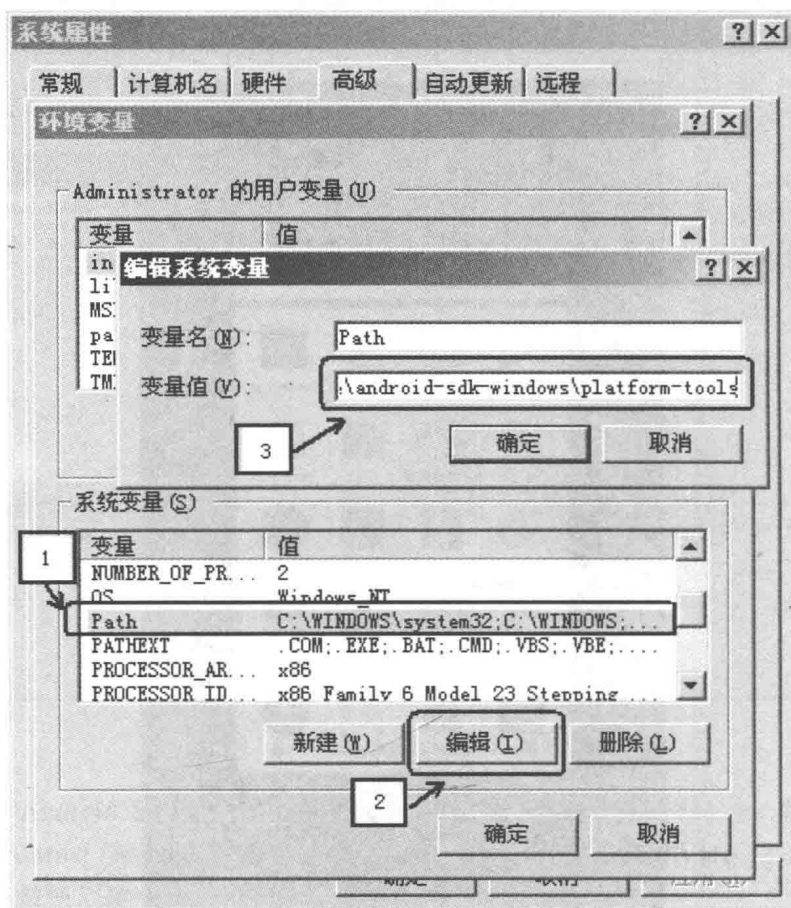


图 1.8 设置 Android 环境变量

1.3 Android API 和在线帮助文档

1. Android API

Android 为用户安装了它所提供的标准类库。所谓标准类库，就是把程序设计所需要的常用方法和接口分类封装成包，Android 提供的标准类库就是 Android API。

在 Android 包中封装了程序设计所需要的主要应用类，本书中所用到的包如下。

- Android.app: 封装了顶层的程序模型，提供基本的运行环境。
- Android.content: 封装了各种对设备上的数据进行访问和发布的类。
- Android.database: 通过内容提供者浏览和操作数据库。
- Android.graphics: 底层的图形库，包含画布、颜色过滤、点、矩形，可以将它们直接绘制到屏幕上。
- Android.location: 封装了定位和相关服务的类。
- Android.media: 封装了一些类管理多种音频、视频的媒体接口。

- **Android.net:** 封装了帮助网络访问的类，超过通常的 `java.net.*` 接口。
- **Android.os:** 封装了系统服务、消息传输、IPC 机制。
- **Android.opengl:** 封装了 `opengl` 的工具、3D 加速。
- **Android.provider:** 封装了类访问 `Android` 的内容提供者。
- **Android.telephony:** 封装了与拨打电话相关的 API 交互。
- **Android.view:** 封装了基础的用户界面接口框架。
- **Android.util:** 涉及工具性的方法，例如时间、日期的操作。
- **Android.webkit:** 默认浏览器操作接口。
- **Android.widget:** 封装了各种 UI 元素（大部分是可见的）在应用程序的屏幕中使用。

2. Android API 帮助文档

Android Studio 提供了离线的 Android API 文档，这是进行程序设计的好工具，希望大家都能用好这个工具。

运行 Android Studio 安装目录下的 `index.html` 文件，运行结果如图 1.9 所示。

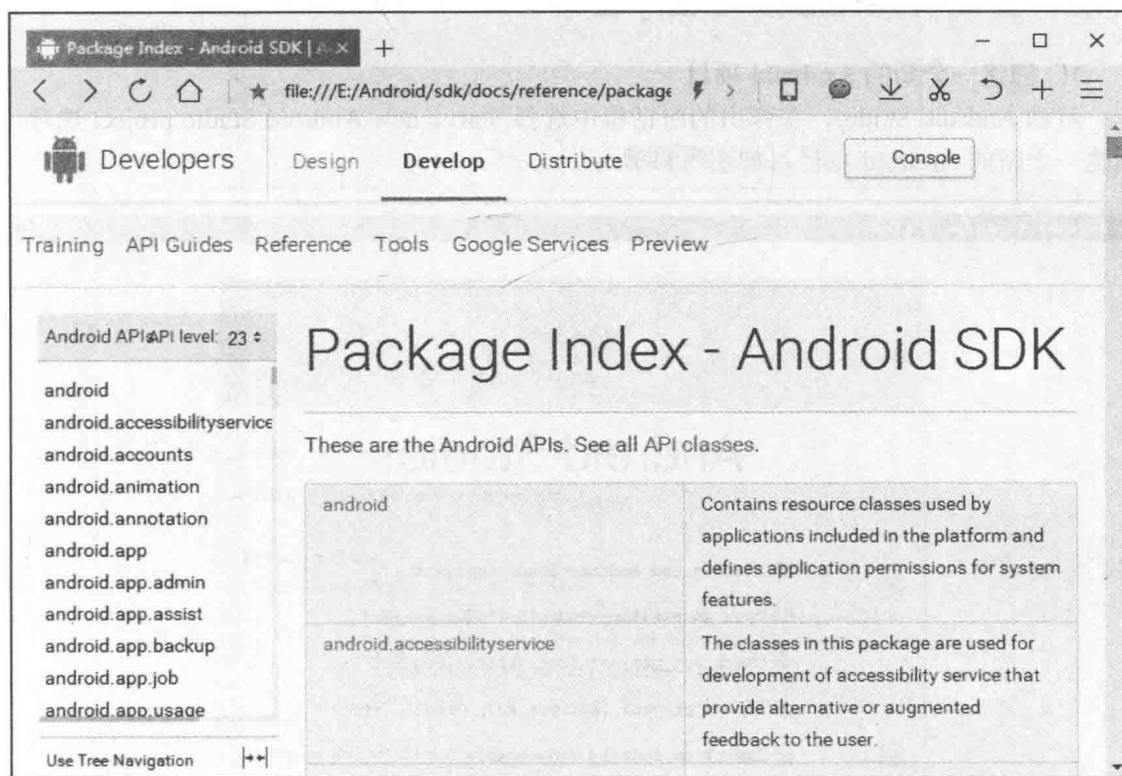


图 1.9 Android API 在线帮助文档

1.4 Android 应用程序的开发过程

1.4.1 开发 Android 应用程序的一般过程

开发 Android 应用程序的一般过程如图 1.10 所示。