

21世纪高等教育计算机规划教材

COMPUTER

Android 移动应用 设计与开发

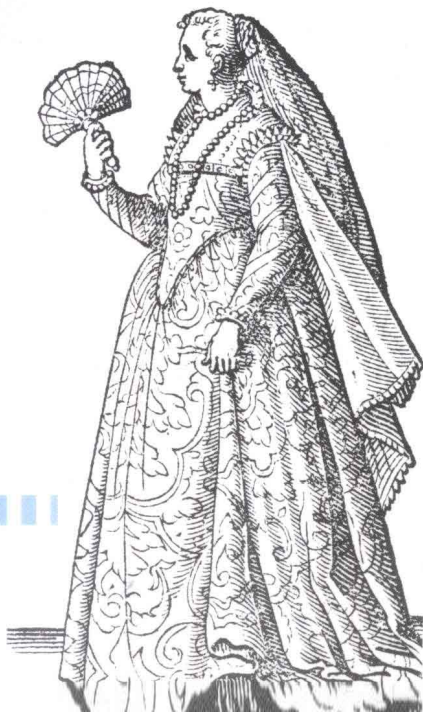
Programming Android

黄宏程 胡敏 陈如松 编著

— 以一个完整案例讲述移动应用开发过程

— 将功能需求穿插到书中各章节

— 力求让读者掌握有效使用 Android 各种组件



 人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等教育计算机规划教材

COMPUTER

Android 移动应用 设计与开发

Programming Android

■ 黄宏程 胡敏 陈如松 编著



人民邮电出版社

北京

图书在版编目 (CIP) 数据

Android移动应用设计与开发 / 黄宏程, 胡敏, 陈如松编著. — 北京: 人民邮电出版社, 2012.9
21世纪高等教育计算机规划教材
ISBN 978-7-115-29040-3

I. ①A… II. ①黄… ②胡… ③陈… 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(2012)第172748号

内 容 提 要

本书以一个完整的案例来讲述移动应用的开发过程,并将其功能需求穿插到书中各章节,通过结合该案例给读者系统而生动地阐述各个知识点,并最终引领读者构建一款完整的移动应用产品。全书共分12章,涵盖Android开发基础知识、基本原理,项目设计和应用发布。内容组织上由易到难,由设计、研发到发布完整的移动应用项目,层层递进,力图使初学者能够快速入门。

本书可作为高等院校理工科各专业本科生相关课程的教材,也可作为开发人员或软件实践者自学和提高的参考书。

21世纪高等教育计算机规划教材 Android 移动应用设计与开发

-
- ◆ 编 著 黄宏程 胡 敏 陈如松
责任编辑 刘 博
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京铭成印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 15.25 2012年9月第1版
字数: 396千字 2012年9月北京第1次印刷

ISBN 978-7-115-29040-3

定价: 32.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前言

Android 是由以 Google 为首的 OHA (Open Handset Alliance) 推出的一款开放的嵌入式操作系统平台。从 2008 年推出 Android SDK 1.0 到现在, Android 系统的市场占有率越来越高, 其应用也是越来越多, 当前正在席卷整个智能手机产业和移动互联网行业。面对这种形势, 很多开发者纷纷转向 Android 应用的开发, 但许多开发者并不熟悉如何在 Android 平台上开发移动互联网应用软件。本书将对 Android 基础知识作系统而详细地讲解, 并通过实际案例让读者了解移动应用产品的开发过程。

目前市场上有着众多讲解 Android 基础开发的书籍, 其中大部分讲述 Android 系统中各种组件的使用。纵观这些书籍, 大都将各个部分分开来进行讲解, 鲜见一本能够使各个部分的内容紧密联系起来的书, 也很少有讲述 Android 客户端如何与移动互联网中的服务器进行交互的书。对于有些初学 Android 编程的读者来说, 如何将各章节所学内容形成一个统一的有机整体开发一个完整的应用, 以及如何让 Android 应用与网络中的服务器进行交互更是充满疑惑。

本书力求让读者掌握如何有效地使用 Android 中的各种组件, 以及 Android 的应用程序是如何与服务器联网和进行数据传递的。本书的特色在于使用一个完整的案例来讲述移动应用的开发过程, 并将其功能需求穿插到书中各章节, 通过该案例给读者系统而生动地阐述各个知识点, 并最终引领读者构建一款完整的移动应用产品。

本书由黄宏程负责内容、架构的确定及各章核心内容撰写, 由重庆邮电大学胡敏副教授和重庆昇通科技有限公司陈如松工程师负责本书主体内容的编著和全书审校。本书共 12 章, 由董子瑜、薛高飞、王蕊、陈思睿、黄宏达、李庆强辅助本书的撰写、案例开发和整理研究工作, 具体如下: 第 1 章由王蕊完成; 第 2、3、12 章由董子瑜完成; 第 5、10 章由黄宏达完成; 第 7、9 章由薛高飞完成; 第 8 章由李庆强完成; 第 4 章由薛高飞和陈思睿共同完成; 第 6 章由董子瑜和李庆强共同完成; 第 11 章和本书的综合案例由黄宏达和李庆强共同完成。

本书核心内容来自于重庆昇通科技有限公司的开发实践和重庆邮电大学的教学总结, 编写时对此做了系统的组织, 并力求做到通俗易懂, 深入浅出。由于作者经验有限, 撰写时间仓促, 书中若有不足之处, 恳请读者批评指正。本书在编著过程中参考了 Android 官网和相关 Android 开发书籍, 在此向相关作者表示诚挚的感谢。

编者

2012 年 6 月

目 录

第 1 章 基础	1
1.1 Android 发展概述	1
1.2 配置开发环境	3
1.2.1 安装 JDK	3
1.2.2 安装 Eclipse	5
1.2.3 安装 SDK	6
1.2.4 安装 ADT	9
本章小结	11
习题	11
第 2 章 Android 项目	12
2.1 第一个应用程序	12
2.1.1 创建 Android 项目	12
2.1.2 项目核心文件和目录	14
2.1.3 开发工具介绍	14
2.2 综合案例	15
2.2.1 需求分析	15
2.2.2 概要设计	17
本章小结	28
习题	28
第 3 章 Android 基本原理	29
3.1 Android 框架	29
3.1.1 体系结构	29
3.1.2 运行原理	30
3.2 Android 应用程序简介	34
3.2.1 Activity 介绍	34
3.2.2 Service 介绍	35
3.2.3 BroadcastReceiver 介绍	36
3.2.4 ContentProvider 介绍	36
本章小结	37
习题	37
第 4 章 Activity 以及 Intent 通信机制	38
4.1 Activity 生命周期	38
4.1.1 Activity 交互机制	38
4.1.2 Activity 状态	38
4.1.3 Activity 生命周期的事件回调方法	39
4.2 Intent	47
4.2.1 Intent 属性	48
4.2.2 Intent 解析	50
4.2.3 Activity 的跳转	51
本章小结	54
习题	54
第 5 章 UI 设计	55
5.1 View	55
5.1.1 TextView 和 EditText	56
5.1.2 Button	57
5.1.3 ImageView	58
5.2 Layout	61
5.2.1 LinearLayout	61
5.2.2 RelativeLayout	65
5.2.3 FrameLayout	70
5.2.4 TableLayout	71
5.2.5 AbsoluteLayout	74
5.3 对话框	74
5.3.1 提示对话框	74
5.3.2 单选对话框	75
5.3.3 复选对话框	76
5.3.4 列表对话框	77
5.3.5 进度条对话框	77
5.3.6 日期选择对话框	78
5.3.7 时间选择对话框	78
5.3.8 拖动对话框	79
5.3.9 自定义对话框	81
5.4 菜单	82
5.4.1 上下文菜单	82

5.4.2 选项菜单	83	8.3 ContentProvider 存储	147
5.4.3 下拉菜单 (Spinner)	85	8.4 文件存储	157
5.5 Toast	87	本章小结	162
5.5.1 常规 Toast	87	习题	162
5.5.2 自定义 Toast	87	第 9 章 网络	163
5.6 Notification	88	9.1 网络访问方式	163
5.6.1 常规 Notification	89	9.1.1 HTTP 通信	163
5.6.2 自定义 Notification	91	9.1.2 Socket 通信	167
5.7 列表	93	9.1.3 HTTP 连接与 Socket 连接区别	169
5.7.1 常用 Adapter	94	9.1.4 案例分析	169
5.7.2 自定义 Adapter	96	9.2 数据解析	173
5.8 AppWidget	99	9.2.1 JSON 格式解析	173
5.9 TabActivity 与 TabHost	104	9.2.2 SAX 方式解析	175
本章小结	107	9.2.3 案例分析	182
习题	107	9.3 获取网络状态	186
第 6 章 服务	109	9.4 JavaScript 与 Java 交互	187
6.1 本地服务	109	9.4.1 WebView	187
6.1.1 两种启动方式	109	9.4.2 在 WebView 中使用 JavaScript	189
6.1.2 生命周期	113	本章小结	192
6.2 远程服务	115	习题	192
6.3 服务小实例	119	第 10 章 重要概念的应用	193
本章小结	125	10.1 进程与线程	193
习题	125	10.1.1 什么是进程	193
第 7 章 广播	126	10.1.2 进程模型	193
7.1 发送和接收广播	126	10.1.3 Android 中进程的生命周期	195
7.1.1 发送广播	126	10.1.4 Android 进程间的通信	196
7.1.2 接收广播	127	10.1.5 什么是线程	196
7.1.3 声明广播	127	10.1.6 Android 的线程模型	196
7.1.4 广播的生命周期	128	10.1.7 进程与线程	197
7.1.5 常见广播 Action 常量	131	10.1.8 进程与线程的重要性	197
7.2 广播小实例	131	10.2 Handler 和 AsyncTask	197
本章小结	133	10.2.1 Handler 的基本概念	197
习题	133	10.2.2 Android 消息机制	197
第 8 章 数据存储和提供者	134	10.2.3 Handler 的具体使用场合	199
8.1 SharedPreferences 存储	134	10.2.4 AsyncTask 的介绍	199
8.2 SQLite 存储	139	10.2.5 AsyncTask 的使用	199
8.2.1 SQLite 简介	139	10.2.6 为什么要在 UI 线程中 创建 AsyncTask 才能使用	200
8.2.2 SQLite 使用	139		

10.2.7 Handler 和 AsyncTask.....	200	习题.....	223
10.3 Application.....	201	第 12 章 发布你的应用程序.....	224
本章小结.....	203	12.1 为应用程序签名.....	224
习题.....	203	12.2 发布和销售应用程序.....	228
第 11 章 综合案例讲解.....	204	12.2.1 在 Google play 上销售你的应用 程序.....	228
11.1 客户端.....	205	12.2.2 其他途径销售你的应用程序.....	230
11.1.1 代码结构.....	205	12.3 更新应用程序.....	230
11.1.2 功能模块.....	211	本章小结.....	234
11.2 服务端.....	220	习题.....	234
11.2.1 总体介绍.....	220	参考文献.....	235
11.2.2 数据解析和控制流程.....	221		
本章小结.....	222		

第 1 章

基础

本章向读者描述 Android 发展过程中几个重要的里程碑和市场数据，带领读者走入 Android 世界；随后，引领读者一步一步配置开发环境，并在配置的过程中向读者讲述各个环节的作用，以使读者较好地明白所做配置的用途。

1.1 Android 发展概述

Android 是 Google 公司在 2007 年 11 月 5 日公布的基于 Linux 平台的开源手机操作系统。早期由 Google 开发，后由开放手机联盟（OHA）开发。自 OHA 成立以后，Android 的发展加快了速度，下面对 Android 的整个发展历程作简要的介绍。

2005 年，Google 收购了成立仅 22 个月的高科技企业 Android，展开了短信、手机检索、定位等业务。2007 年 11 月 5 日，Google 宣布推出基于 Linux 平台的开源手机操作系统，正式命名为 Android，该平台由操作系统、中间件、用户界面和应用软件组成，Google 对外宣称其是首个为移动终端打造的真正开放和完整的移动软件，同月，Google 宣布组建一个全球性的联盟组织——开放手机联盟（Open Handset Alliance，OHA）。这一联盟支持 Google 发布的 Android 手机操作系统或者应用软件，与 Google 共同开发 Android 这一开放源代码的移动操作系统。开放手机联盟包括电信运营商、半导体芯片商、手机硬件制造商、软件厂商和商品化公司等 5 类。

谷歌 Android SDK 1.0 是 Android 系统最早的版本，发布自 2008 年 9 月。同年，美国运营商 T-Mobile 在纽约正式发布第一款 Android 手机——T-Mobile G1（见图 1.1），并且开放了其源代码，任何人或机构都可以免费使用 Android，同时 Android 的源代码可以从 Android Open Source Project 网站下载，网址为 <http://source.android.com/index.html>。



图 1.1 T-Mobile G1

Android 各个版本的市场份额占有比在 2012 年 2 月的统计结果如图 1.2 所示，从此图中我们可以看出 Android 操作系统的版本很多，每个版本的市场占有率不均等。同月数据显示，全球智能手机出货量中，智能手机的各操作系统平台的份额构成如图 1.3 所示，Android 或基于 Android 定制的操作系统占 17.6%，iPhone 占 4.4%，Symbian 占 4.3%，BlackBerry 占 2.76%，Bada 占 1%，Windows Phone 占 0.5%，Android 依然处于领先地位。

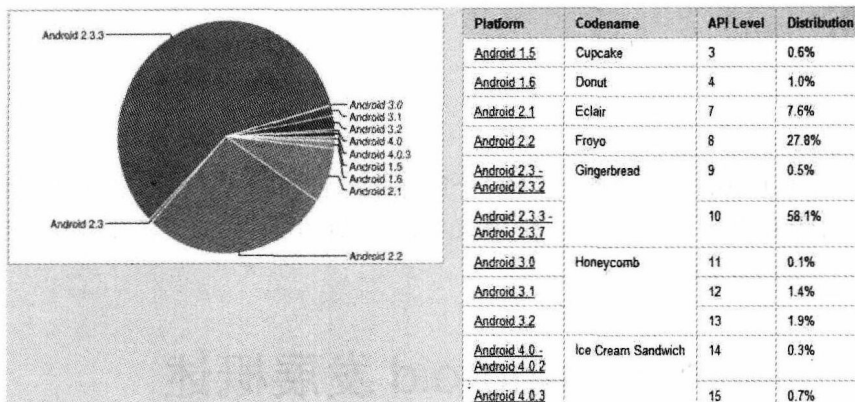


图 1.2 Android SDK 各个版本的市场份额占有比

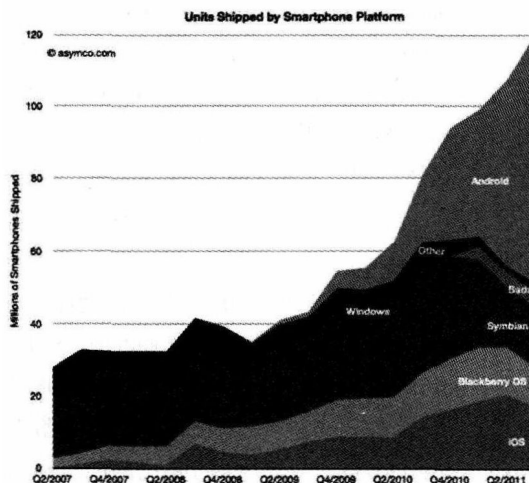


图 1.3 2012 年全球智能手机市场份额

随着 3G/WIFI 技术的成熟、智能手机的普及以及基于智能手机的各种应用和服务的增长，移动互联网在真正意义上进入了高速发展的阶段，而移动互联网的迅速发展也得益于统一的软件平台和移动终端硬件的发展。随着 IT 技术的不断发展，电子产品功能越来越智能化，嵌入式操作系统也得到了迅猛的发展，但是由于产品功能、应用场合等不同也造成不同操作系统百家争鸣的状态，而由 Google 统领的 OHA 开放手机联盟共同推出的开源软件平台将一统混乱的局面，为各个产业链提供一个完美的公共开发平台。Android 手机等移动终端追求高质量的用户体验，在用户体验上有了全新的改变，同时终端的发展也使得硬件上得到了显著的升级。

1.2 配置开发环境

本节讲述 Windows 操作系统下的开发环境的配置，在配置之前需要做好相应的准备，包括相关安装包的选择和下载。本节将按照安装的实际情景进行截图说明，并讲述在安装过程中会遇到的常见问题和解决方法。

1.2.1 安装 JDK

开发 Android 应用程序的时候，仅有 Java 运行环境（Java Runtime Environment, JRE）是不足以开发 Android 应用程序的，需要完整的 JDK（JDK 包含了 JRE），且要求其版本在 1.5 以上，可选择 JDK1.5 或 JDK1.6，可以访问网站 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 进行下载，下载页面如图 1.4 所示，当前最新版本为 Java SE 7u3。

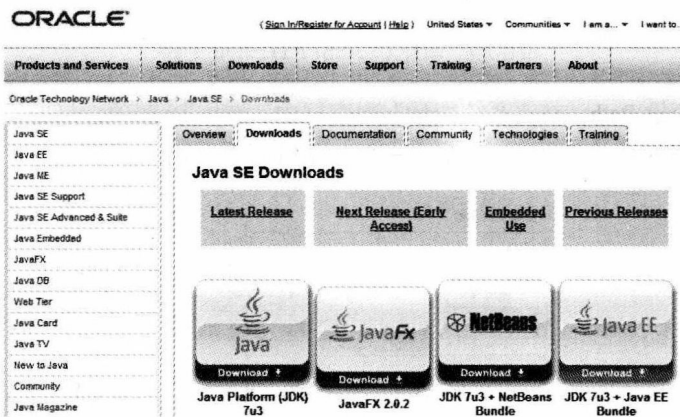


图 1.4 JDK 下载页面

安装 J2SDK 以后，若需要在 cmd 下使用 Java 命令和编译、运行程序，可以配置环境变量：新建环境变量 JAVA_HOME，右键单击“我的电脑”→“属性”→“高级”→“环境变量”，如图 1.5 所示。

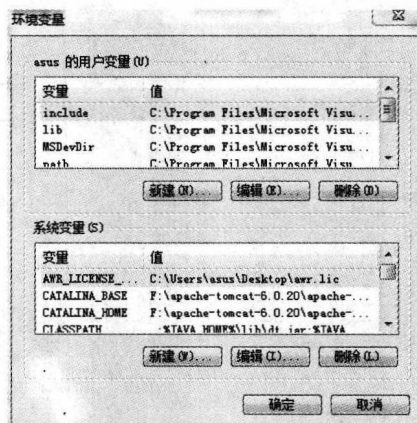


图 1.5 “环境变量”对话框

(1) 单击系统变量下的“新建”按钮，变量名为 JAVA_HOME，变量值为 E:\Program Files\code\Java\jdk1.5 (JDK 的安装路径)，然后确认，如图 1.6 所示。

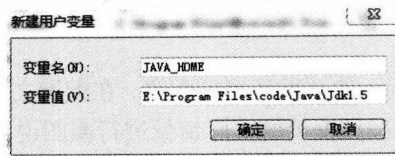


图 1.6 新建 JAVA_HOME 环境变量

(2) 新建环境变量 CLASSPATH (步骤和上一步类似)，变量名为 CLASSPATH，变量值为 %JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;，然后确认，如图 1.7 所示。

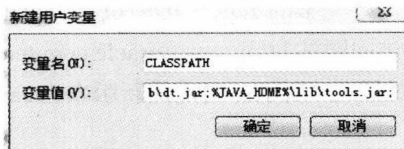


图 1.7 新建 CLASSPATH 环境变量

(3) 编辑环境变量 Path，在“系统变量”列表框中，选中 Path 项，单击下面的“编辑”按钮，在“变量值”文本框的最前面加入 %JAVA_HOME%\bin；如图 1.8 所示。

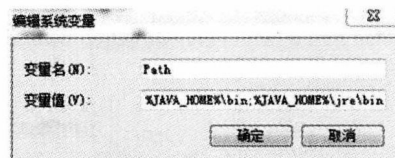


图 1.8 编辑系统变量

单击“确定”按钮，完成环境变量配置。

配置环境变量后，单击“开始”→“运行”输入 cmd，在弹出的 DOS 窗口中输入 javac 回车，得到如图 1.9 所示结果，说明配置成功。

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

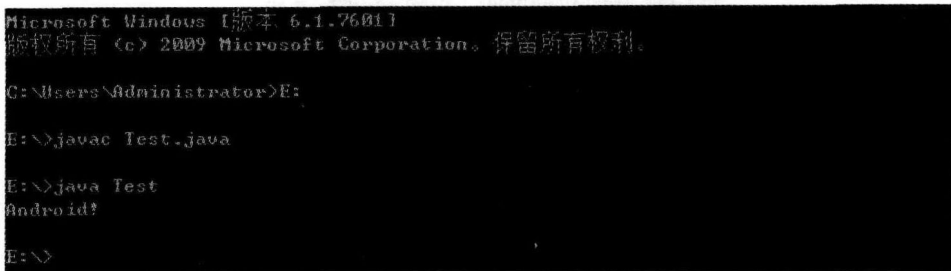
C:\Users\Administrator>javac
用法: javac <选项> <源文件>
其中, 可能的选项包括:
  -h
    生成所有调试信息
  -g
    生成某些调试信息
  -nowarn
    不生成任何警告
  -verbose
    输出所有编译正在执行的选项的消息
  -deprecation
    输出使用已过时的 API 的源位置
  -classpath <路径>
    指定要检索用户类文件和注释处理程序的位置
  -cp <路径>
    指定要检索用户类文件和注释处理程序的位置
  -sourcepath <路径>
    指定要检索源代码文件的位置
  -bootclasspath <路径>
    指定要检索启动类库的位置
  -extdirs <目录>
    指定要检索扩展类库的位置
  -endorseddirs <目录>
    指定要检索已签名的类库的位置
  -proc {none|only}
    控制是否执行在源代码和/或编译时
    的字节码验证
  -processor {<class1>|<class2>|...|<classN>}
    要执行的注释处理程序的名称。通过默认
    的类搜索路径
  -processorpath <路径>
    指定要检索注释处理程序的位置
  -d <目录>
    指定存放生成的类文件的位置
  -s <目录>
    指定存放生成的源文件的位置
  -implicit {<none>|<class>}
    指定是否隐式引用了生成类文件
```

图 1.9 运行结果

也可以用一个小程序测试一下，代码如下：

```
public class Test
{
public static void main(String args[])
{
System.out.println("Android! ");
}
}
```

将以上代码保存为一个 Test.java 文件，假设保存在 E:\，单击“开始”→“运行”（输入 cmd），在弹出的 dos 窗口中输入 E:（回车），继续输入 javac Test.java 回车，最后输入 java Test 回车，如图 1.10 所示则说明运行完毕，JDK 配置成功。



```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>E:

E:\>javac Test.java

E:\>java Test
Android!

E:\>
```

图 1.10 运行结果

1.2.2 安装 Eclipse

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。Eclipse 附带了一个标准的插件集，包括 Java 开发工具（Java Development Tools, JDT），可以用 Eclipse 集成开发环境（Integrated Development Enviroment, IDE）进行 Android 开发。安装好 JDK 后，就可以接着安装 Eclipse 了，可以访问网站 <http://www.eclipse.org/downloads/> 下载，下载页面如图 1.11 所示。



图 1.11 Eclipse 的下载页面

选择 Windows 平台的 Eclipse IDE for Java EE Developers 下载，下载完成后，找到下载的压缩包 eclipse-java-indigo-SR2-win32.zip，Eclipse 无须安装程序，解压此压缩文件到目标文件即可，进

入解压后的目录,找到 Eclipse 可执行文件,然后在桌面上创建一个快捷方式,双击此快捷方式直接运行,Eclipse 能自动找到用户安装的 JDK 路径,启动界面如图 1.12 所示(注意:Eclipse 压缩包存放路径不要含汉字,以免打开可执行文件时出错)。

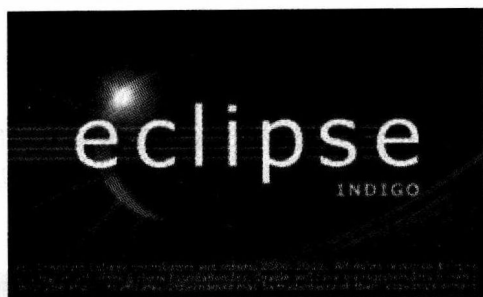


图 1.12 启动 Eclipse 界面

1.2.3 安装 SDK

Android SDK(Software Development Kit)提供了在 Windows/Linux/Mac 平台上开发 Android 应用的开发组件,Android 支持所有的平台,其包含了在 Android 平台上开发移动应用的各种工具集。Android SDK 不仅包括了 Android 模拟器和用于 Eclipse 的 Android 开发工具插件(ADT),而且包括了各种用来调试、打包以及在模拟器上安装应用的工具。Android SDK 主要是以 Java 语言为基础,用户可以使用 Java 语言来开发 Android 平台上的软件应用。通过 SDK 提供的一些工具将其打包成 Android 平台使用的 apk 文件,然后用 SDK 中的模拟器(Emulator)来模拟和测试软件在 Android 平台上运行情况 and 效果。

安装好 JDK 和 Eclipse 后,下一步需要下载安装 Android SDK,以便进行 Android 应用程序的开发,可以通过访问 Android developers 网站(网址为 <http://developer.android.com/sdk/index.html>)下载 Android SDK,下载页面如图 1.13 所示。

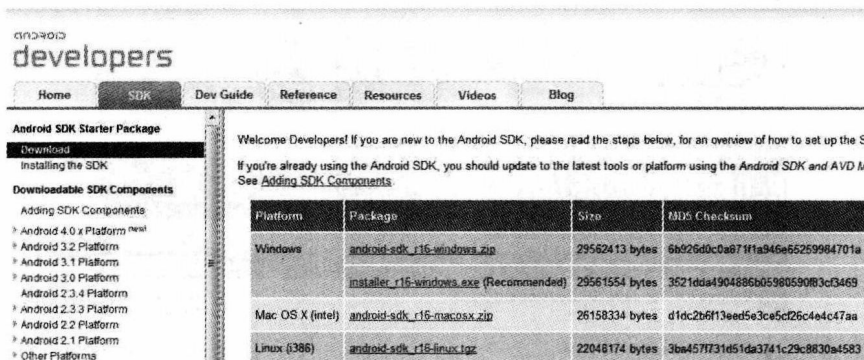


图 1.13 Android SDK 下载页面

如 Android SDK 下载页面所示,不同的操作系统需下载不同的 Android SDK 压缩包,这里选择 Windows 平台的 android-sdk_r16-windows.zip 下载。下载完后,即可安装 Android SDK,现有两种安装方式:通过 eclipse sdk manager 在线安装和离线安装。

1. 通过 eclipse sdk manager 在线安装

首先解压缩 android-sdk_r16-windows.zip,并双击运行 eclipse sdk manager.exe,如图 1.14 所示。

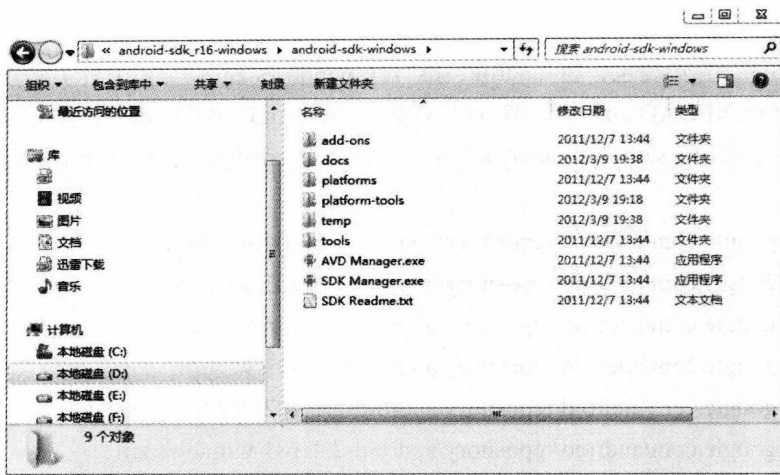


图 1.14 android-sdk_r16-windows 目录结构

接着可以购选想要安装的 Package 进行安装，如图 1.15 所示。

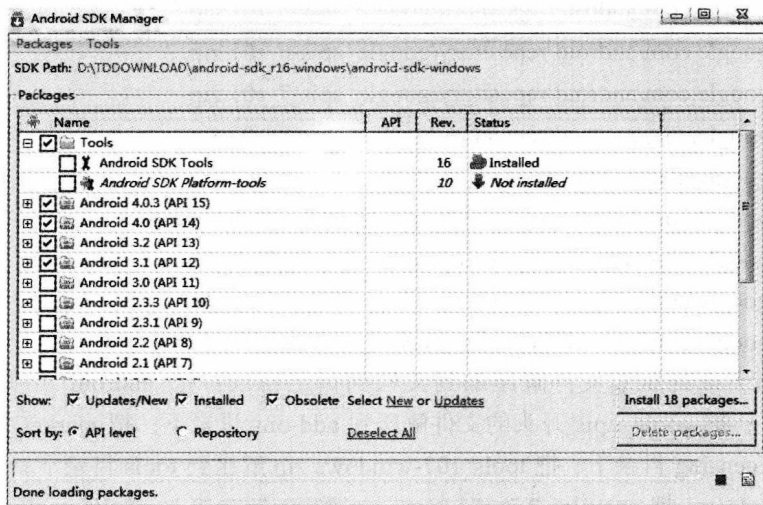


图 1.15 Android SDK Manager 页面

确认对话框如图 1.16 所示，选择 Accept All，单击“Install”按钮进行下载安装，即可完成 Android SDK 的安装。

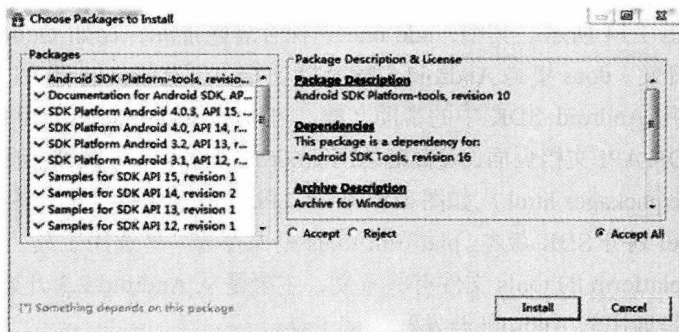


图 1.16 Choose Packages to Install 页面

2. 离线安装

首先下载 SDK 的安装包, 将 android-sdk_r16-windows.zip (安装工具) 解压到目录, 如我的目录 D:\TDDOWNLOAD\android-sdk_r16-windows\android-sdk-windows, 然后新建以下几个文件夹: platforms、docs、samples、usb_driver、market_licensing。接着打开下载工具, 下载以下内容:

http://dl-ssl.google.com/android/repository/android-1.5_r04-windows.zip
http://dl-ssl.google.com/android/repository/android-1.6_r03-windows.zip
http://dl-ssl.google.com/android/repository/android-2.0.1_r01-windows.zip
http://dl-ssl.google.com/android/repository/android-2.0_r01-windows.zip
http://dl-ssl.google.com/android/repository/android-2.1_r02-windows.zip
http://dl-ssl.google.com/android/repository/android-2.2_r01-windows.zip
http://dl-ssl.google.com/android/repository/docs-2.2_r01-linux.zip
http://dl-ssl.google.com/android/repository/google_apis-3-r03.zip
http://dl-ssl.google.com/android/repository/google_apis-4_r02.zip
http://dl-ssl.google.com/android/repository/google_apis-5_r01.zip
http://dl-ssl.google.com/android/repository/google_apis-6_r01.zip
http://dl-ssl.google.com/android/repository/google_apis-7_r01.zip
http://dl-ssl.google.com/android/repository/google_apis-8_r01.zip
<http://dl-ssl.google.com/android/repository/jboss-5.1.0.GA.zip>
http://dl-ssl.google.com/android/repository/samples-2.1_r01-linux.zip
http://dl-ssl.google.com/android/repository/samples-2.2_r01-linux.zip
http://dl-ssl.google.com/android/repository/tools_r06-windows.zip
http://dl-ssl.google.com/android/repository/usb_driver_r03-windows.zip

最后把上面下载的压缩文件解压缩到文档到指定目录: 把 android 开头的文件解压到 platforms 目录下; 把 google_apis 开头的文件解压到 add-ons 目录下; 把 market_licensing-r01.zip 解压到 market_licensing 目录下; 把 tools_r07-windows.zip 解压到 tools 目录下; 把 docs-2.2_r01-linux.zip 解压到 docs; 把 samples-2.2_r01-linux.zip 和 samples-2.1_r01-linux.zip 解压到 samples 目录下; 把 usb_driver_r03-windows.zip 解压到 usb_driver 目录下; 就基本完成了 Android SDK 的安装工作, 再打开 SDK Manager.exe 发现, 它会找到我们已经安装的内容, 就完成了整个 Android SDK 安装工作。

在 Android SDK 安装到本地磁盘后, 可以在文件系统中查看到 Android SDK 的结构。Android SDK 的目录结构如图 1.14 所示。其中, add-one 里保存着附加库, 比如 Google Map, 支持基于 Google Map 的地图开发。docs 里是 Android SDK API 文档, 所有的 API 都可以在这里查到, 开发 Android 应该学会使用 Android SDK 中的帮助文档, 可以在 docs\reference\目录下面的 index.html 文件打开 Android SDK API 文档页面, 或直接访问 Android developers 网站(地址: <http://developer.android.com/reference/packages.html>), 如图 1.17 所示。Platforms 是每个平台的 SDK 真正的文件, 里面会根据 API Level 划分 SDK 版本。platform-tools 里保存着一些通用工具, 比如 adb、aapt、aidl、dx 等工具, 这里和 platform 的 tools 文件有些重复, 主要是从 Android 2.3 开始这些工具被划分为通用的了。Tools 下是通用的 Android 开发和调试工具。

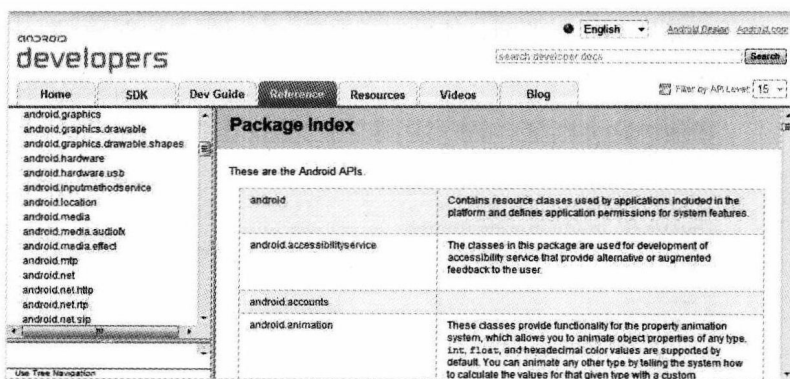


图 1.17 Android SDK API 文档

1.2.4 安装 ADT

Android 为 Eclipse 订制了一个插件，即 Android Development Tools (ADT)，为 Android 开发提供开发工具的升级或者变更，也为用户提供一个强大的综合环境用于开发 Android 应用程序。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，以及用 SDK 工具集调试应用程序。

现在开始在线安装 Eclipse ADT 插件，首先启动 Eclipse，选择 Help→Install New Software, 打开 Eclipse 的插件安装页面，如图 1.18 所示。

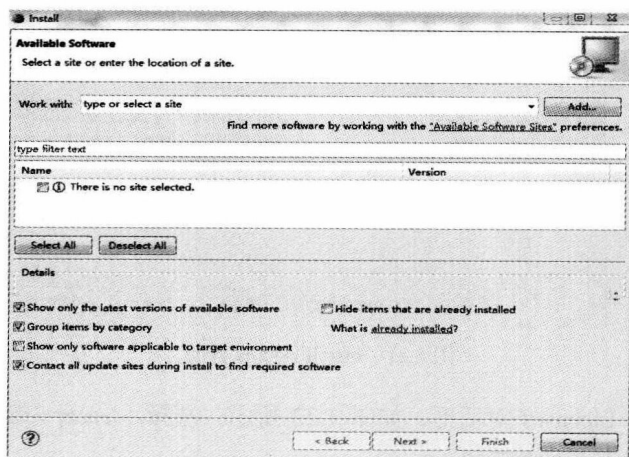


图 1.18 Eclipse 的插件安装界面

单击 Add 按钮，弹出 Add Repository 对话框，如图 1-19 所示，在 Name 文本框中输入 ADT，在 Location 文本框中输入 <http://dl-ssl.google.com/android/eclipse/>，单击 OK 按钮后，Eclipse 会自动搜索可安装的插件。

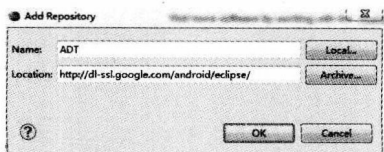


图 1.19 Add Repository 界面

此时 Eclipse 会搜索指定 URI 的资源，如果搜索无误，会出现 Developer Tools 复选框，如图 1.20 所示，选中复选框，单击 Next 按钮。

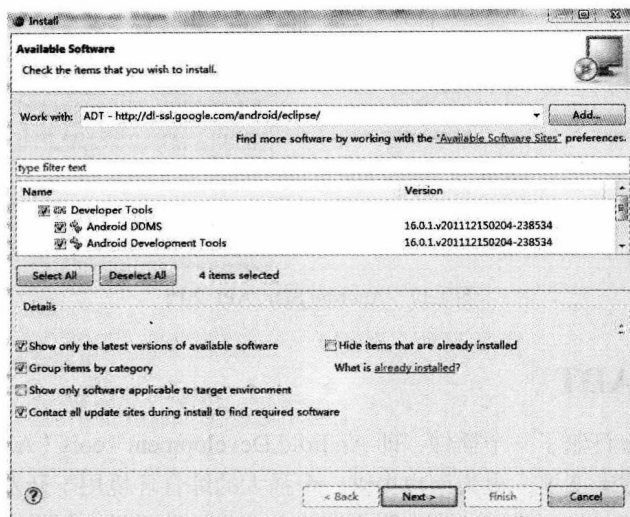


图 1.20 Eclipse 的插件安装界面

接着出现 Install Details 对话框，如图 1.21 所示，单击 Next 按钮。

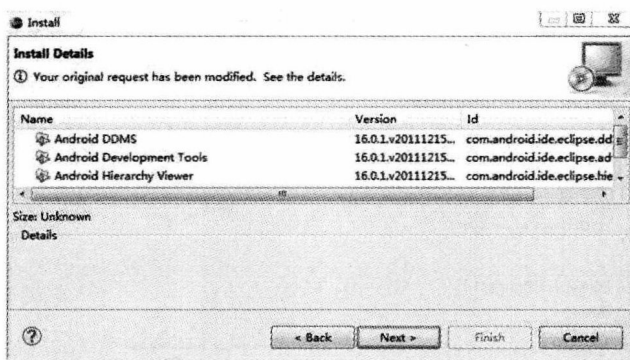


图 1.21 Install Details 界面

然后出现 Review Licenses 对话框，如图 1.22 所示，选择 I accept the terms of the license agreements 复选框，单击 Finish 按钮。

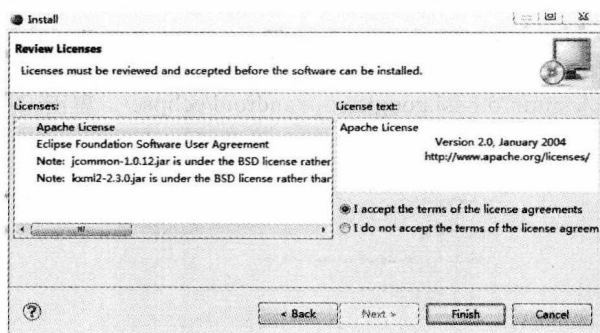


图 1.22 Review Licenses 界面