

21世纪高等院校移动开发人才培养规划教材
21Shiji Gaodeng Yuanxiao Yidong Kaifa Rencai Peiyang Guihua Jiaocai

Android项目开发 入门教程

张伟华 主编



Android Inventor Tutorial

4个典型项目

任务分解，逐层完成

由浅入深，循序教学



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

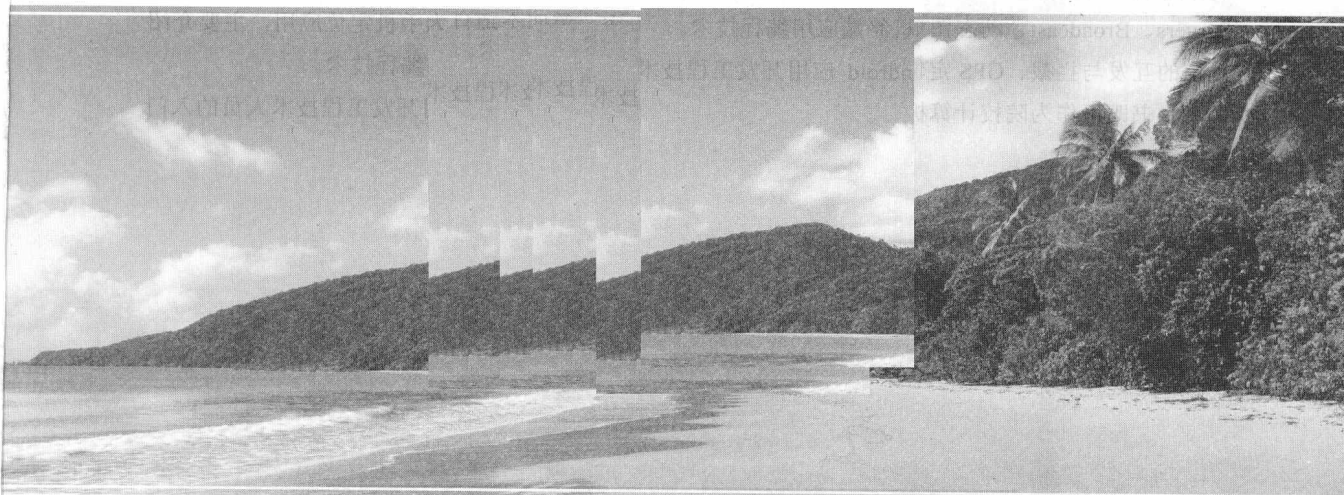
21世纪高等院校移动开发人才培养

21Shiji Gaodeng Yuan

rcal Peiyang

Android项目开发 入门教程

张伟华 主编 朱东 伊雯雯 副主编 李金友 高乐 参编



Android Inventor Tutorial

人民邮电出版社

北京

图书在版编目(CIP)数据 820300310

Android项目开发入门教程 / 张伟华主编. — 北京 :
人民邮电出版社, 2015.12
21世纪高等院校移动开发人才培养规划教材
ISBN 978-7-115-37927-6

I. ①A… II. ①张… 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(2015)第123814号

内 容 提 要

本书精选4个典型项目贯穿全书,通过任务分解的方式,逐步学习各个知识点。第一个项目为开发准备,主要介绍Android开发环境搭建和Android程序的创建步骤。第二个项目为“我爱记单词”,融合了移动商务所需的各项技术,主要介绍界面设计、网络互连、数据管理等内容。第三个项目为数独游戏,主要介绍游戏编程,包括游戏编程所需的技术如图形用户界面、资源、Intent、Activity、Servers、Broadcast、动态图形绘制、音频文件控制等内容。第四个项目为手机定位应用,主要介绍短信的互发与拦截、GPS定位与地图编程、传感器的编程等手机终端应用编程技术。

本书既可作为院校计算机专业的教科书,又可作为从事Android应用开发工程技术人员的入门参考书。

-
- ◆ 主 编 张伟华
 - 副 主 编 朱 东 伊雯雯
 - 参 编 李金友 高 乐
 - 责任编辑 范博涛
 - 责任印制 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京鑫正大印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
 - 印张: 14.75 2015年12月第1版
 - 字数: 368千字 2015年12月北京第1次印刷
-

定价: 35.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316
反盗版热线: (010)81055315

前言 PREFACE

中国已经是全球手机用户最多的国家,随着 3G 和 4G 网络的广泛使用,移动互联网的迅猛发展,智能手机日益普及,手机移动开发已成为一个高速发展的领域。Android 作为一个开放的平台,适用各种移动终端,用户数量日益增长。基于 Android 技术的软件开发应用前景非常广阔,高校学生及广大软件开发者认识到这个趋势,踊跃加入到学习 Android 的行列中来。因此,编写一本易于教学和学习的教材显得非常必要。

目前,市面上 Android 技术的书籍对于知识的介绍一般都是按章节来阐述,案例的处理一般都只针对一个知识点来展开介绍,缺乏对一个完整项目实施过程的介绍。当前教育普遍推行基于工作过程的任务驱动教学模式,市场上这类教材比较少,难以满足教学需求。本书基于 Android 学习和教学的需要进行编写,按照任务驱动的方式组织学习内容,按照应用需求分类将 Android 主要应用技术依托 4 个项目进行介绍,每个项目以任务的方式分解完成,在任务介绍过程中,依次按照任务目标、任务内容、任务分析、实现过程、技术要点、技能训练、任务拓展七部分展开。在任务实施过程中加强动手实践的能力,在技术要点中介绍任务中所需要的技术,并在技能训练和任务拓展环节辅以任务训练,强化对内容的理解。

本书由具有多年从事 Java、Android 技术开发和教学经验的教师编写而成。苏州工业职业技术学院的张伟华老师任主编,朱东、伊雯雯老师任副主编,企业工程师李金友、高乐对书的撰写提供了许多案例和宝贵意见,在此表示衷心感谢!

由于编者水平有限,书中难免存在疏漏和不足,恳请同行专家和读者能给予批评和指正。

编 者

2015 年 9 月

目 录 CONTENTS

项目一 开发准备 1

任务 1.1 开发环境搭建	2	任务描述	12
任务描述	2	任务目标	12
任务目标	2	任务分析	12
任务分析	2	知识要点	13
知识要点	3	任务实现	21
任务实现	6	技能训练	26
任务 1.2 第一个 Android 应用程序开发	12		

项目二 我爱记单词 27

任务 2.1 用户登录界面设计	29	知识要点	65
任务描述	29	任务实现	67
任务目标	29	技能训练	71
任务分析	29	任务 2.5 单词存取	71
知识要点	30	任务描述	71
任务实现	37	任务目标	71
技能训练	43	任务分析	72
任务 2.2 系统退出功能设计	44	知识要点	72
任务描述	44	任务实现	75
任务目标	45	技能训练	83
任务分析	45	任务 2.6 单词共享	85
知识要点	45	任务描述	85
任务实现	47	任务目标	85
技能训练	49	知识要点	86
任务 2.3 用户注册界面设计	49	任务分析	90
任务描述	49	任务实现	90
任务目标	50	拓展学习	98
任务分析	50	任务 2.7 用户信息网络传输	99
知识要点	51	任务描述	99
任务实现	56	任务目标	99
任务 2.4 用户访问信息存取	64	任务分析	99
任务描述	64	知识要点	100
任务目标	65	任务实现	102
任务分析	65	技能训练	113

任务 2.8 单词网络下载	114	任务分析	115
任务描述	114	知识要点	115
任务目标	115	任务实现	116

项目三 数独游戏 131

任务 3.1 数独游戏界面设计	132	任务分析	149
任务描述	132	知识要点	149
任务目标	133	任务实现	149
任务分析	133	任务 3.4 游戏背景音乐设置	156
知识要点	133	任务描述	156
任务实现	133	任务目标	156
任务 3.2 九宫格界面绘制	140	任务分析	156
任务描述	140	知识要点	157
任务目标	140	任务实现	158
任务分析	141	任务拓展	162
知识要点	141	任务 3.5 继续游戏功能实现	166
任务实现	142	任务描述	166
技能训练	147	任务目标	167
任务 3.3 数字键盘设计与实现	148	任务分析	167
任务描述	148	知识要点	167
任务目标	148	任务实现	168

项目四 手机定位应用 174

任务 4.1 界面与数据层设计	175	知识要点	185
任务描述	175	任务实现	189
任务目标	176	任务拓展	195
任务分析	176	任务 4.3 地图显示联系人位置	203
任务实现	176	任务描述	203
技能训练	184	任务目标	203
任务 4.2 短信发送与接收处理	184	任务分析	204
任务描述	184	知识要点	204
任务目标	184	任务实现	208
任务分析	184	任务拓展	219

项目情境

“工欲善其事必先利其器”，在开始 Android 开发之旅之前，要先搭建开发环境。作为一个 Android 应用程序开发人员，掌握 Android 开发环境的配置是必须的，只有掌握了最基本的环境配置，才能进行后续的项目开发。在本项目中我们要完成 Android 应用程序开发环境搭建，然后完成第一个 Android 应用程序。

学习目标

- ☑ 能够下载 Android 的开发工具包。
- ☑ 能够搭建 Android 开发环境。
- ☑ 能够创建 Android 应用程序。
- ☑ 掌握 Android 应用程序框架。

工作任务

任务名称
任务 1.1 开发环境搭建
任务 1.2 第一个 Android 应用程序开发

任务 1.1 开发环境搭建

任务描述

在进行 Android 开发之前，需要搭建相应的开发环境。本任务主要实现 Android 开发环境的搭建，包括 JDK 的安装与配置、ADT 和 AVD 的配置、Android SDK 的配置。

名词解析

Java 开发工具包 (Java Development Kit, JDK)：是 Java 开发所必需的开发包。

Android 开发工具 (Android Developer Tools, ADT)：是集成在 Eclipse 上开发调试 Android 应用程序的工具。

Eclipse：是基于 Java 的可扩展开发平台。Android 推荐使用 Eclipse 来开发 Android 应用，并为它提供了专门的插件 ADT。

Android 软件开发工具包 (Android Software Development Kit, Android SDK)：是 Android 专属的软件开发工具包。

Android 系统模拟运行设备 (Android Virtual Device, AVD)：用于在电脑上模拟手机进行 Android 应用程序运行的设备。

Java 应用程序接口 (Application Programming Interface, API)：是 Java 开发工具中预先定义的函数，为软件开发人员提供访问某软件或硬件例程代码。

任务目标

- ① 了解 Android 的历史和版本。
- ② 了解 Android 的系统架构。
- ③ 知道如何下载 Android 开发所需的工具包。
- ④ 能安装并配置 JDK。
- ⑤ 能配置 ADT，并能使用 ADT 进行 Android 应用程序开发。
- ⑥ 能配置 Android SDK，并能进行 Android SDK 的升级管理。
- ⑦ 会配置 AVD，并能使用 AVD 进行 Android 应用程序运行调试。

任务分析

本书所用的 Android 开发环境的工具包主要包括 JDK、Android SDK、Eclipse 和 ADT 等软件包。这些开发工具包都可以通过网络下载，具体实现过程：

- ① 下载工具包；
- ② 安装配置 JDK；
- ③ 安装配置 ADT；
- ④ 配置 Android SDK；

⑤ 配置并启动 AVD。

知识要点

1. Android 的由来

Android 是一种基于 Linux 内核的开放源代码的操作系统，主要应用于移动设备，如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发。在中国较多人将其翻译为“安卓”。

2003 年，Andy Rubin 等人创建 Android 公司；2005 年 Google 公司收购 Android 公司后，继续开发运营 Android 系统；2008 年 Google 公司推出了 Android 的最早版本 Android 1.0；2009 年 Google 公司推出了 Android 1.5。从这个版本开始，Android 的后续版本均用一个甜品的名称来命名。随着后续发展，越来越多的“甜品”（Android 版本）被 Google 公司陆续推出，让我们来认识一下这些“甜品”吧。

下面依次介绍 Android 版本。

2009 年 4 月发布的 Android 1.5: Cupcake（纸杯蛋糕）。

2009 年 9 月发布的 Android 1.6: Donut（甜甜圈）。

2010 年 1 月发布的 Android 2.1: Éclair（巧克力泡芙）。

2010 年 5 月发布的 Android 2.2 : Froyo（冷冻酸奶）。

2010 年 12 月发布的 Android 2.3: Gingerbread（姜饼）。

2011 年 2 月发布的 Android 3.0/3.1: Honeycomb（蜂巢）。

2011 年 10 月发布的 Android 4.0: Ice Cream Sandwich（冰淇淋三明治）。

2012 年 6 月发布的 Android 4.1: Jelly Bean（果冻豆）。

2013 年 9 月发布的 Android 4.4: KitKat（奇巧巧克力）。

2014 年 6 月 Google 公司发布了最新的 Android L，即 Android5.0 系统。

各版本的 Logo 如图 1-1 所示。



图 1-1 Android 版本

2. Android 系统架构

Android 系统架构从软件分层角度来看,可分为应用程序层 (Application)、应用框架层 (Application Framework)、系统库 (Libraies)、Android 运行时 (Android Runtime)、Linux 内核层 (Linux Kernel) 5 个部分。Android 系统架构如图 1-2 所示。

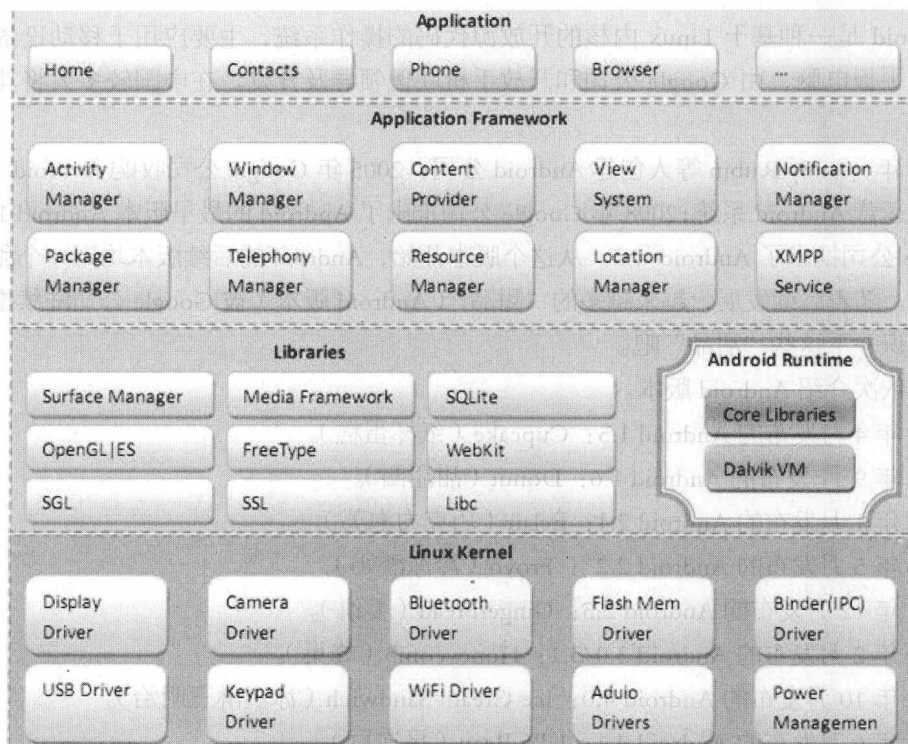


图 1-2 Android 系统架构

(1) 应用程序层

应用程序层 (Application) 包含在 Android 设备上运行的所有应用,它们是 Android 系统中直接面向用户的部分。

Android 平台不仅仅是操作系统,也包含了许多应用程序,如 SMS 短信客户端程序、电话拨号程序、图片浏览器、Web 浏览器等应用程序。这些应用程序是用 Java 语言编写的,且这些应用程序都是可以其他应用程序所替换,不同于其他手机操作系统固化在系统内部的系统软件,它们更加灵活和个性化。

(2) 应用框架层

应用框架层 (Application Framework) 是 Android 开发的基础,很多核心应用程序是通过这一层来实现其核心功能的。该层简化了组件的重用,开发人员可以直接使用其提供的组件来进行快速的应用程序开发,也可以通过继承而实现个性化的拓展。

- 活动管理器 (Activity Manager): 管理各个应用程序生命周期及通常的导航回退功能。
- 窗口管理器 (Window Manager): 管理所有的窗口程序。

- 内容提供者 (Content Provider): 使得不同应用程序之间存取或者分享数据。
- 视图系统 (View System): 构建应用程序的基本组件。
- 通告管理器 (Notification Manager): 使得应用程序可以在状态栏中显示自定义的提示信息。
- 包管理器 (Package Manager) Android: 系统内的程序管理。
- 电话管理器 (Telephony Manager): 管理所有的移动设备功能。
- 资源管理器 (Resource Manager): 提供应用程序使用的各种非代码资源, 如本地化字符串、图片、布局文件、颜色文件等。
- 位置管理器 (Location Manager): 提供位置服务。
- XMPP 服务 (XMPP Service): 提供 Google Talk 服务。

(3) 系统库

系统库 (Libraies) 是指一些提供底层功能支持的库 (主要由 C/C++ 编写的), 是连接应用程序框架层与 Linux 内核层的重要纽带。

系统库各个部分的功能如下。

- Surface Manager: 执行多个应用程序时候, 负责管理显示与存取操作间的互动, 另外也负责 2D 绘图与 3D 绘图进行显示合成。
- 多媒体库 (Media Framework): 基于 PacketVideoOpenCore; 支持多种常用的音频、视频格式录制和回放, 编码格式包括 MPEG4、MP3、H.264、AAC、ARM。
- SQLite: 小型的关系型数据库引擎。
- OpenGL|ES: 根据 OpenGL ES 1.0 API 标准实现的 3D 绘图函数库。
- FreeType: 提供点阵字与向量字的描绘与显示。
- WebKit: 网页浏览器的软件引擎。
- SGL: 底层的 2D 图形渲染引擎。
- SSL: 在 Andorid 上通信过程中实现握手。
- Libc: 从 BSD 继承来的标准 C 系统函数库, 专门为基于 embedded linux 的设备定制。

(4) Android 运行时

Android 运行时 (Android Runtime) 的应用程序是采用 Java 语言编写, 但它并不使用 J2ME 来执行程序, 而是在 Android 自带的 Android 运行时中执行。Android 运行时分为 Java 核心库和 Dalvik 虚拟机两部分。

Java 核心库: Java 核心库提供了 Java 应用程序接口 (Application Programming Interface, API) 中的大多数功能, 同时也包含了 Android 的一些核心 API, 如 android.os、android.net、android.media 等。

Dalvik 虚拟机: Android 程序不同于 J2ME 程序, 每个 Android 应用程序都有一个专有的进程, 并且不是多个程序运行在一个虚拟机中, 而是每个 Android 程序都有一个 Dalvik 虚拟机的实例, 并在该实例中执行。Dalvik 虚拟机是一种基于寄存器 (register-based) 的虚拟机, 不是传统的基于栈 (stack-based) 的虚拟机, 进行了内存资源使用优化及支持多个虚拟机的特点。

需要注意的是: 不同于 J2ME, Android 程序在虚拟机中执行的并非编译后的字节码, 而

是通过转换工具 dx 将 Java 字节码转成 dex 格式的中间码。

(5) Linux 内核层

Linux 内核层 (Linux Kernel) 主要指 Android 寄宿的 Linux 操作系统及相关驱动。通常来说, 只有硬件厂商和从事 Android 移植的开发者, 才会基于此来进行开发。Android 是基于 Linux 2.6 内核, 其核心系统服务如安全性、内存管理、进程管理、网路协议以及驱动模型都依赖于 Linux 内核。

任务实现

1. 下载工具包

Android 开发环境搭建之前需要在网上下载以下开发工具包, 做好环境配置的准备工作。

(1) JDK 工具包的下载

在进行 Android 开发之前, 需要下载安装 Java 的开发环境, 因为 Android SDK 的应用层是 Java 语言, 所以需要 Java 开发环境。有过 Java 开发经验的读者, 对 JDK 的配置应该并不陌生。本书使用的 JDK 是 32 位的 JDK6。

JDK 下载地址是 http://www.java.net/download/jdk6/6u10/promoted/b32/binaries/jdk-6u10-rc2-bin-b32-windows-i586-p-12_sep_2008.exe

(2) 下载 ADT 集成开发工具

<http://wear.techbrood.com/sdk/index.html> 为 Android 在国内的镜像网址。最新的 Android 开发工具和都会在该网址发布, 也是 Android 学习者必须了解的网站。目前该网站提供了一个集成各种必备插件的 ADT Bundle 下载, 包括 Eclipse、Android SDK 和 ADT 等系列软件包。打开 <http://wear.techbrood.com/sdk/index.html>, 如图 1-3 所示, 可下载 Android 开发工具。

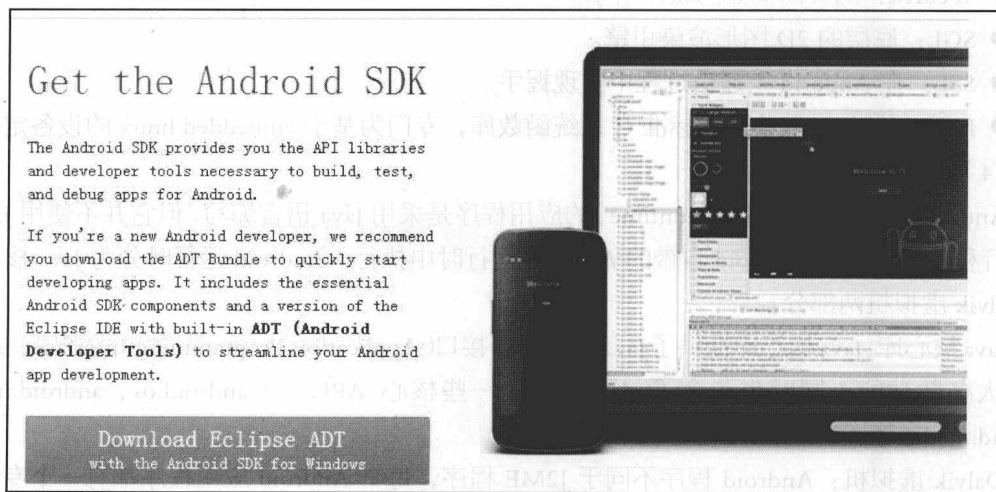


图 1-3 Android 开发工具下载

2. 安装配置 JDK

JDK 开发工具包是 Java 和 Android 应用程序开发的必备工具。从网上下载 JDK 工具包之后进行安装, 其安装步骤较简单, 基本按照默认设置即可。现有的 Java 开发书籍中对 JDK 的

安装都有介绍，此处不再赘述。

安装好 JDK 后，需要进行 JDK 环境变量配置。右键单击“我的电脑”→“属性”→“高级”→“环境变量”→“系统变量”。系统变量设置需要设置 3 个环境变量，其中 JAVA_HOME 的内容为 JDK 的安装路径，ClassPath 指向 Java 的类，Path 指向 Java 的编译、运行等命令工具。具体配置如下，等号左侧的是环境变量名，等号右侧的是环境变量值。

- JAVA_HOME= C:\Program Files\Java\jdk1.7.0_09
- ClassPath=.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;
- Path=%JAVA_HOME%\bin;

注意：其中点“.”和分号“;”不能缺少，而且必须是英文。

3. 安装配置 Eclipse 和 ADT

在 Android 的早期开发中，需要配置 Eclipse、Android SDK 和 ADT。但是现在已经方便很多，可以直接从 Android 的官网下载开发包。该开发包已经将 Android 开发工具集成在一个压缩包中，名为 Android adt-bundle-windows-x86。压缩包中包含 Eclipse+ADT、Android SDK、AVD 和 Android 系统平台。安装配置 Eclipse 和 ADT 步骤如下。

① 解压网上下载的“adt-bundle-windows-x86”压缩包。其中包含已经配置 ADT 的 eclipse3.8 工具包和 Android SDK 文件夹。解压后文件结构如图 1-4 所示。本书使用的是 Android4.2 版本。可以利用 SDK 文件夹下的 SDK Manager 下载需要的 Android 版本。

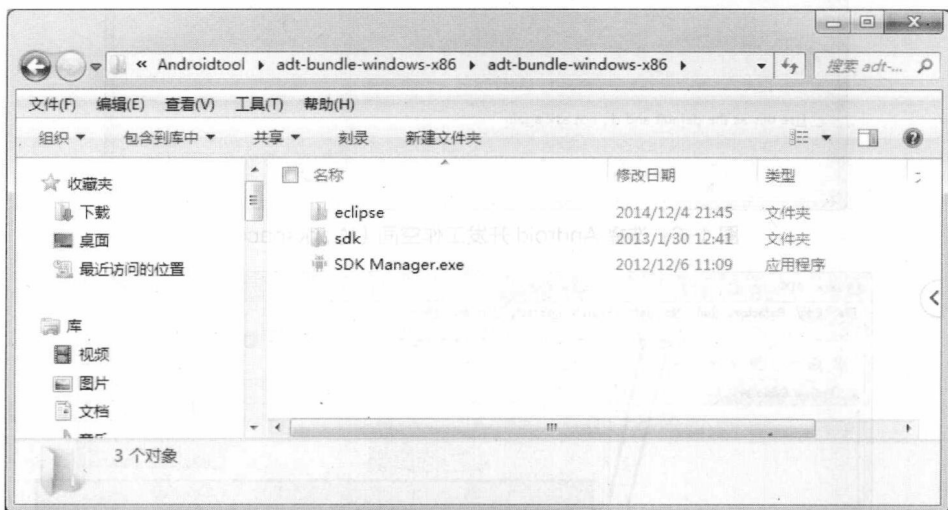


图 1-4 adt-bundle-windows-x86 文件夹结构

② 启动 ADT，因为 ADT 已经集成在 Eclipse 中，所以启动 ADT 的步骤如下。

- 打开 eclipse 文件夹，如图 1-5 所示，鼠标左键双击 eclipse 程序。
- 设置工作空间 (workspace) 如图 1-6 所示。进入 ADT 后的界面如图 1-7 所示，进入该界面之后就可以进行 Android 应用程序开发了。具体开发过程将在下一个任务中介绍。

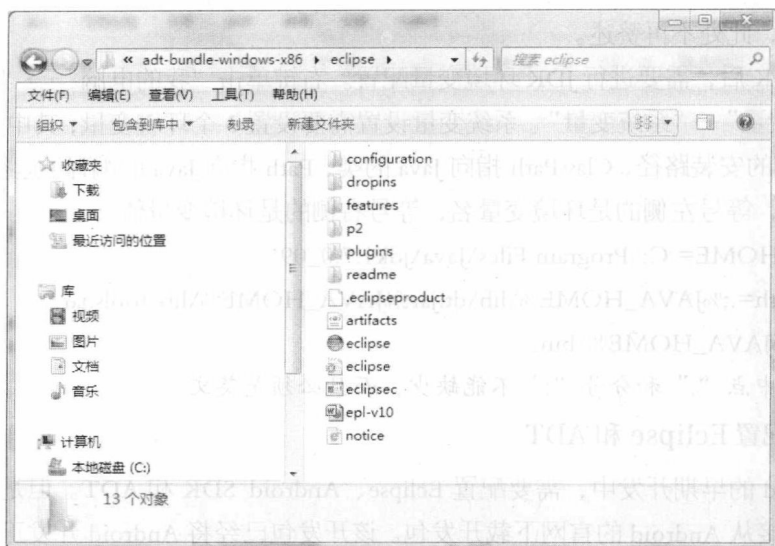


图 1-5 eclipse 文件夹结构

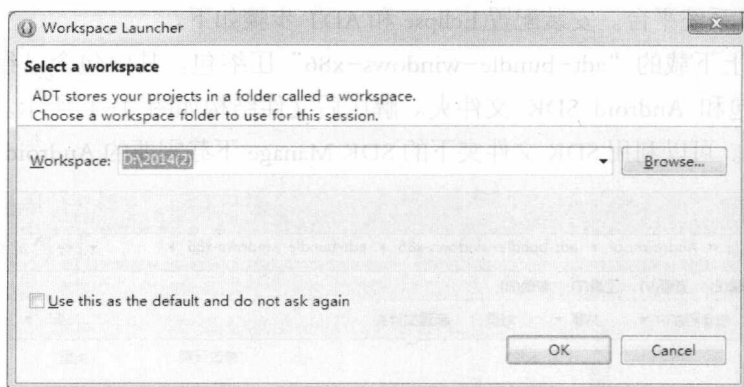


图 1-6 选择 Android 开发工作空间 (Workspace)

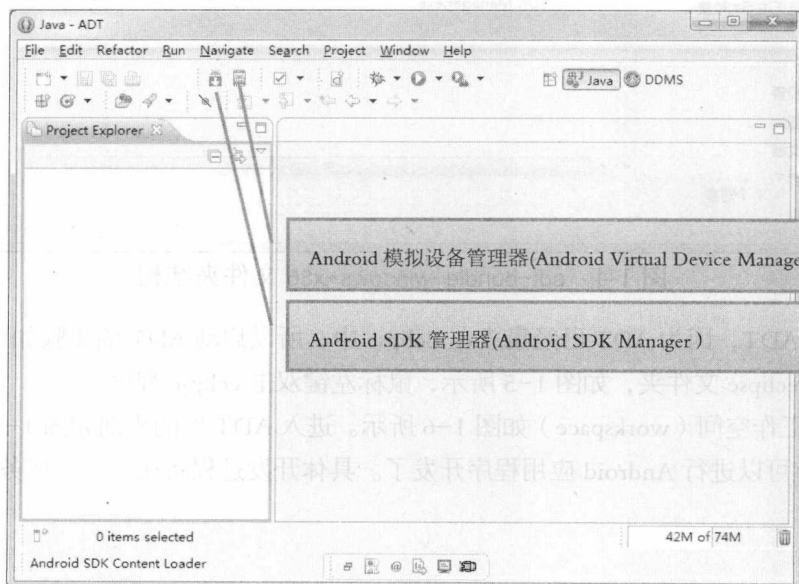


图 1-7 ADT 开发环境

4. 配置 Android SDK

Android SDK 是用于支持 Android 开发的软件工具包,adt-bundle-windows-x86 中已经包含 Android 4.2 的 SDK。一般系统会自动获取 Android SDK 的路径,如果系统未自动获取则需手动配置。为了便于后续的开发工作,这里将讲述如何配置 Android SDK 的路径及如何升级 Android SDK。

(1) 配置 Android SDK 路径

如果 ADT 中没有自动配置 Android SDK 路径,则需要手动配置 Android SDK 路径。其步骤为,打开“ADT”→“选择 Windows”→选择“Preferences”选项,如图 1-8 所示。在弹出的图 1-9 所示界面中选中“Android”属性,并在 SDK Location 输入框中选择之前解压缩的 Android SDK 目录→单击 Apply 按钮,再单击 OK 按钮。

提示: SDK Location 为解压 adt-bundle-windows-x86 压缩包后 adt-bundle-windows-x86 中 Android SDK 文件夹的路径。

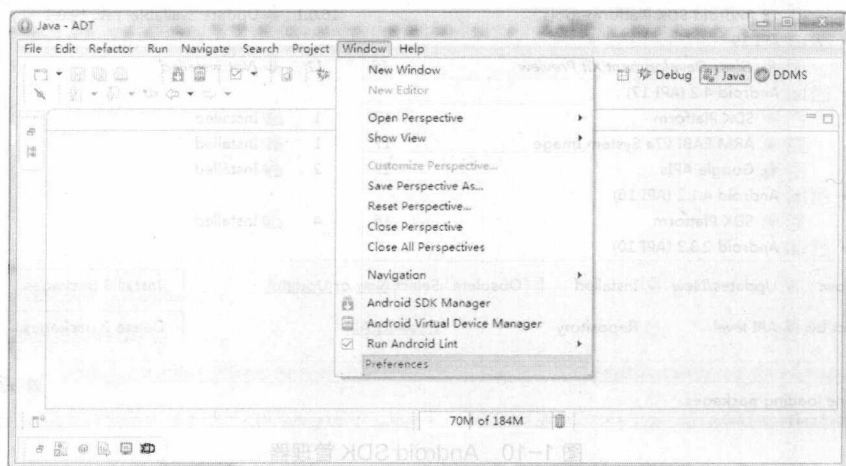


图 1-8 选择 Windows 的 Preferences 项

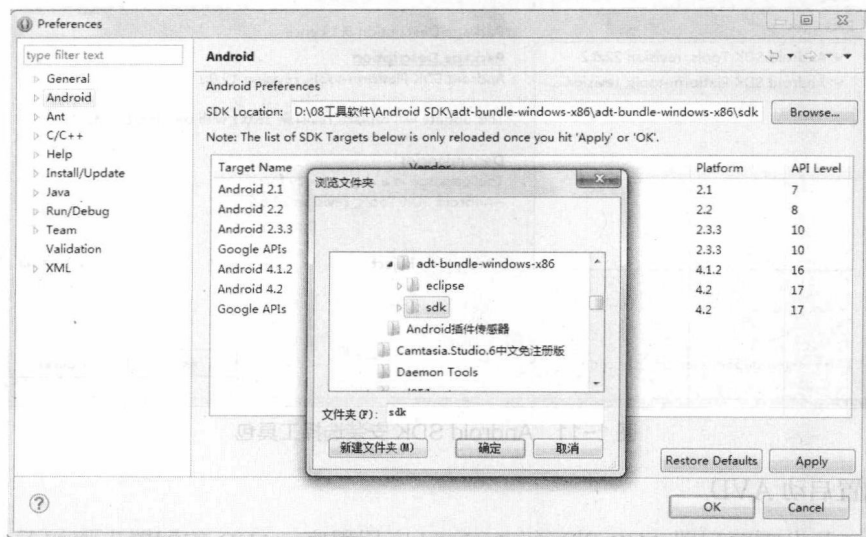


图 1-9 Android SDK 设置

(2) 升级 Android SDK

单击 ADT 集成开发环境主页面的“Android SDK Manager”项，如果网络连接正常可用的话，“Android SDK Manager”窗体中将列出当前已经安装的 Android SDK 软件和其他可用版本的 Android SDK 列表，如图 1-10 所示。

选择要安装的选项，单击“Install3packages”，弹出图 1-11 所示界面，选择“Accept All”。此时就会通过网络从 Google 服务器下载这些软件包。视所选软件包的多少和网速快慢，下载时间可能耗费数小时之久，数据量为 2~5GB。

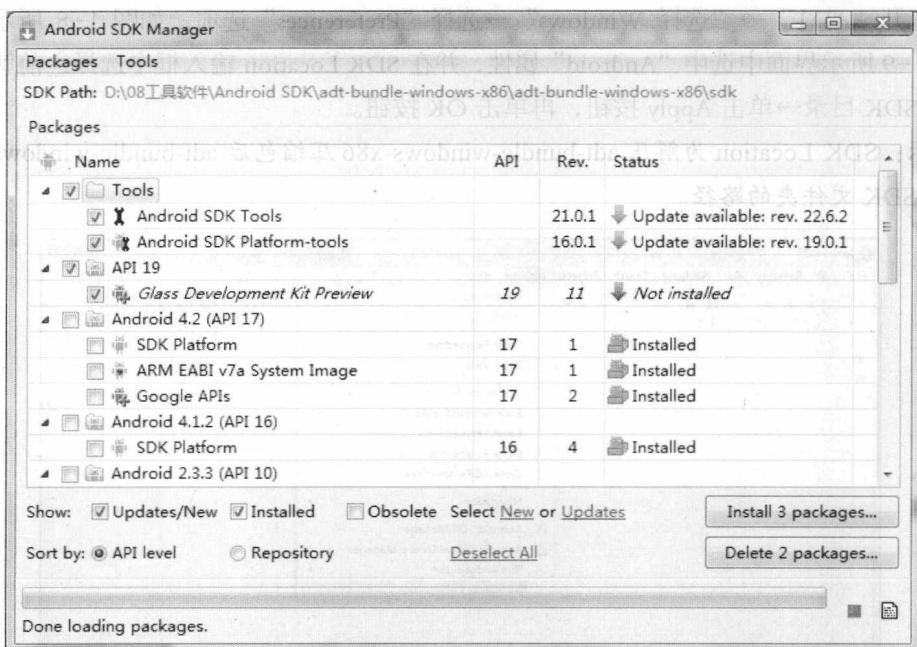


图 1-10 Android SDK 管理器

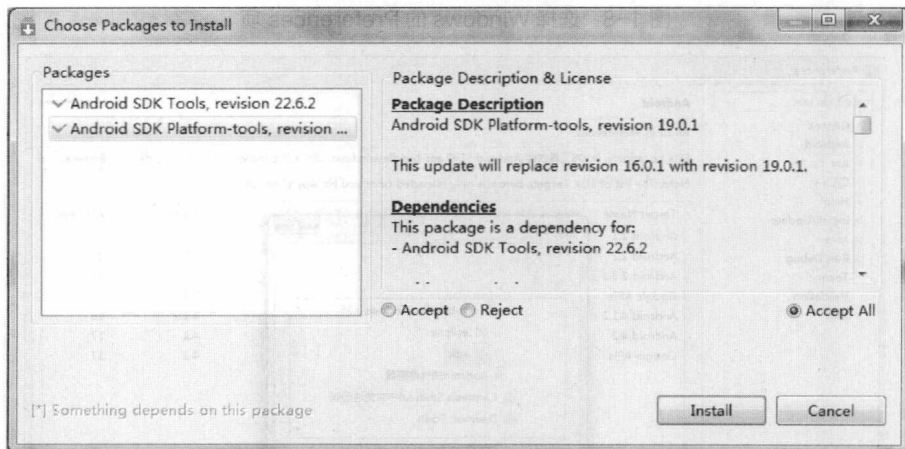


图 1-11 Android SDK 安装选择工具包

5. 配置启动 AVD

Android 开发需要配置 AVD 以运行 Android 应用程序。AVD 的配置步骤如下。

① 在打开的 ADT 中选择  按钮，弹出界面如图 1-12 所示，单击 New 按钮开始对 AVD

进行配置,如图 1-13 所示,其配置参数如下。

- AVDName: 按照规范命名,此处命名为 android4.2。
- Target: Android 4.2- API Level17。

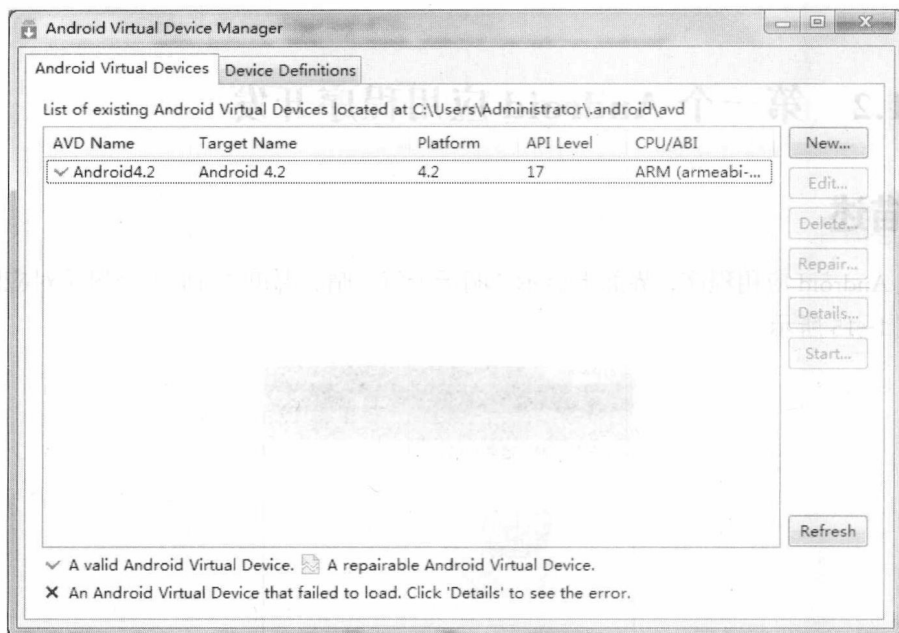


图 1-12 Android AVD 设备管理

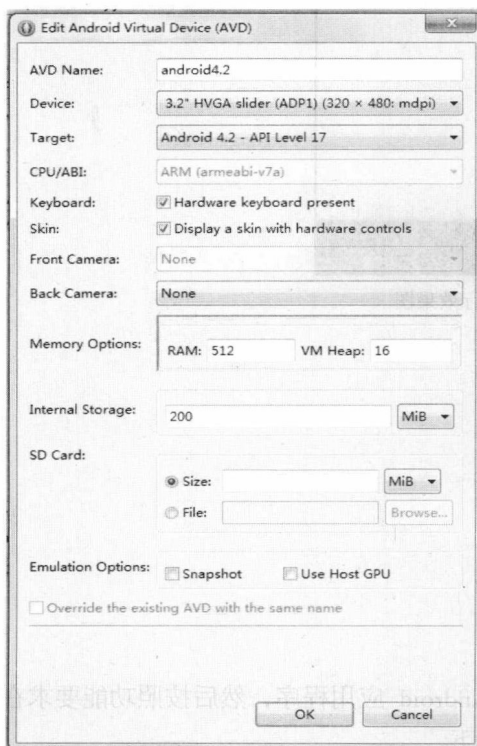


图 1-13 Android AVD 设置图



图 1-14 AVD 运行效果图