

高等学校“十三五”规划教材·计算机软件工程系列

Mobile Application Development for Android Device

# Android 移动设备应用程序开发

◎主 编 马 超 张亚楠 王 姚



哈尔滨工业大学出版社  
HARBIN INSTITUTE OF TECHNOLOGY PRESS

# Android 移动设备应用程序开发

主 编 马 超 张亚楠 王 姚

哈爾濱工業大學出版社

## 内 容 简 介

本书以 Android 移动设备的应用程序开发为主题,以 Android 6.0 官方 API 文档为基础,详细介绍了 Android 移动设备应用程序开发涉及的各个环节,从基础开发、UI 设计、网络通信技术、数据持久化、传感器开发等多个角度,为读者提高程序设计能力提供参考与帮助。

全书 12 章:包括 Android 系统的发展史,Android 基础程序开发,结合 Material Design 设计思想的移动应用 UI 设计,Android 数据存储,百度地图 API 应用,Android 网络通信开发,Android 近距离通信开发,Android 传感器开发等内容。

本书可作为相关专业高年级本科生教学用书,也可作为 Android 移动设备应用程序开发人员的参考书。

### 图书在版编目(CIP)数据

Android 移动设备应用程序开发/马超,张亚楠,王姚  
主编. —哈尔滨:哈尔滨工业大学出版社,2016.6

ISBN 978 - 7 - 5603 - 5854 - 3

I. ①A… II. ①马… ②张… ③王… 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(2016)第 173732 号

责任编辑 孙连嵩 刘 威

出版发行 哈尔滨工业大学出版社

社 址 哈尔滨市南岗区复华四道街 10 号 邮编 150006

传 真 0451 - 86414749

网 址 <http://hitpress.hit.edu.cn>

印 刷 哈尔滨市石桥印务有限公司

开 本 787mm × 1092mm 1/16 印张 18.25 字数 467 千字

版 次 2016 年 6 月第 1 版 2016 年 6 月第 1 次印刷

书 号 ISBN 978 - 7 - 5603 - 5854 - 3

定 价 68.00 元

---

(如因印装质量问题影响阅读,我社负责调换)

# 前 言

Android 是由 Google 公司和开放手机联盟共同开发的一款基于 Linux 内核的开源代码的操作系统,主要应用在移动设备上。例如:智能手机、平板电脑、数字电视以及可穿戴设备等。截止至 2015 年的第四季度,Android 平台手机的全球市场份额已经达到 80.75%,苹果的 iOS 系统占据了 17.75% 的份额,在 iOS 和 Android 以外,其他的系统如 Windows Phone 和黑莓,两者分别仅有 1.1% 和 0.2% 的占有率。随着黑莓放弃自家平台,转投 Android 阵营,以及 Windows Phone 的业绩持续不振,Windows Phone 和黑莓的市场占有率很可能会持续下滑。目前,全球知名手机品牌多采用 Android 系统,例如:三星、华为、联想、小米、HTC 等,而采用 iOS 系统的手机仅有苹果一家。

对于学习 Java 编程语言的读者,Android 操作系统的出现,提供新的学习方向,巨大的市场需求,提供更多的就业机会,也急需更多的开发者来提供更加丰富的应用。本书主要针对学习过 Java 编程语言,具备一定的编程基础,有意愿学习 Android 平台应用程序开发的读者。多数学习程序开发的读者在熟悉了语法知识之后,都迫不及待地想编写一款属于自己的软件,这是值得肯定的学习编程的积极态度。但是,如果所选择的项目过大、过于复杂,往往很难实现其功能。所以,对于 Android 应用程序开发的初学者,建议选择功能单一、结构简单的案例项目。

本书从学习 Android 的实用性以及技术热点的角度出发,充分考虑了 Android 初学者在进行移动设备应用程序开发时所需要掌握的基础知识,其内容共分为 12 章。第 1 章介绍 Android 应用程序开发的基础知识;第 2 章介绍如何在 Android 中创建一个最简单的 Hello World 应用程序;第 3 章介绍 Android UI 设计语言的基础知识;第 4 章介绍 Android 界面设计基础;第 5 章介绍 Android 中的活动组件、意图对象以及广播接收器组件;第 6 章介绍 Android 后台 Service 组件;第 7 章介绍 Android 数据库与存储技术;第 8 章介绍 Android 位置服务与地图应用;第 9 章介绍 Android 多线程;第 10 章介绍 Android 网络通信开发;第 11 章介绍 Android 近距离通信开发;第 12 章介绍 Android 传感器开发。每章

都有相应的实例,以便读者更好地理解该章的内容。

本书在编写过程中,参考了 Android 6.0 官方 API 文档,按照知识的逻辑关系来编写,循序渐进、突出重点,对知识点的讲解与介绍尽可能做到全面、准确,并给出知识点的适用场合。对于重点、难点知识,采取给出易于理解的案例项目,按步骤讲解实现的方式。全书所有章节讲解知识的方式统一,结构清晰,方便读者快速查询相关问题。在介绍章节内容时,根据不同知识点的具体情况,介绍知识点的分类、周边信息并总结功能实现的步骤。

本书由马超、张亚楠、王姚编写。由于作者学术与经验的局限,在本书的结构、知识点与难点的选择和解析过程中,难免会存在一定的问题与不足,希望广大读者不吝赐教。相关技术问题可以发送邮件到 [machao8396@163.com](mailto:machao8396@163.com) 进行交流,作者会尽量给予答复。

编者

2016年5月



# 目 录

第1章	Android 简介 .....	1
1.1	Android 发展史 .....	1
1.2	Android 平台架构及特性 .....	3
1.3	开发环境的搭建 .....	5
第2章	HelloWorld Android 应用程序 .....	15
2.1	Eclipse + ADT 环境 .....	15
2.2	Android Studio 环境 .....	20
第3章	UI 设计语言 Material Design .....	29
3.1	Material Design 简介 .....	29
3.2	Material Design 的环境 .....	30
3.3	Material Design 的特性 .....	31
3.4	Material Design 的高度和阴影 .....	32
3.5	Material Design 的布局设计基础 .....	42
第4章	Android 界面设计基础 .....	60
4.1	Android 布局设计 .....	60
4.2	Android 控件设计 .....	69
第5章	活动、意图与广播 .....	114
5.1	活动与碎片 .....	114
5.2	使用 Intent 链接 Activity .....	121
5.3	Intent 过滤器 .....	131
5.4	广播与 BroadcastReceiver .....	134
第6章	Android 后台服务 .....	138
6.1	Service .....	138
6.2	Service 启动方式比较 .....	149
6.3	AIDL .....	155
第7章	Android 数据存储 .....	163
7.1	SharedPreferences 数据存储 .....	163
7.2	内部存储 .....	167

# 目 录

7.3	外部存储 .....	169
7.4	数据库存储 .....	175
7.5	网络存储 .....	189
7.6	数据共享 .....	192
<b>第 8 章</b>	<b>Android 位置服务与地图应用 .....</b>	<b>202</b>
8.1	位置服务 .....	202
8.2	百度地图应用 .....	208
<b>第 9 章</b>	<b>Android 多线程 .....</b>	<b>218</b>
9.1	Android 下的线程 .....	218
9.2	循环者 - 消息机制 .....	218
9.3	AsyncTask .....	223
<b>第 10 章</b>	<b>Android 网络通信开发 .....</b>	<b>228</b>
10.1	HTTP 网络通信 .....	229
10.2	Socket 网络通信 .....	235
10.3	URL 通信 .....	240
10.4	WiFi 管理 .....	243
<b>第 11 章</b>	<b>Android 近距离通信开发 .....</b>	<b>247</b>
11.1	蓝牙 Bluetooth .....	247
11.2	近场通信 NFC .....	257
<b>第 12 章</b>	<b>Android 传感器开发 .....</b>	<b>274</b>
12.1	Sensor 开发基础 .....	274
12.2	Sensor 应用实例 .....	277
<b>参考文献</b>	.....	<b>286</b>

# 第 1 章 Android 简介

Android 一词的本义是指“机器人”，同时也是 Google 于 2007 年 11 月宣布的基于 Linux 平台的开源手机操作系统的名称。2008 年 9 月，德国的移动运营商 T-Mobile 公司在美国纽约举办了一场大型的新品发布会，并利用这个机会向所有智能手机爱好者隆重介绍了全世界第一款基于 Android 平台的智能手机 G1。

T-Mobile G1 是来自我国台湾省的 HTC (High Tech Computer) 宏达公司定制，其内部研发代号为 Dream (中文含义：梦想)。Android 平台由 Google 发起的“开放手持设备联盟”开发，因此称 T-Mobile G1/HTC Dream 是 Google 手机。目前，Android 逐渐扩展到平板电脑及其他领域上，例如：电视、数码相机、游戏机以及各种智能穿戴设备等。

## 1.1 Android 发展史

2003 年 10 月，Andy Rubin 等人创建 Android 公司，并组建 Android 团队。2005 年 8 月，Google 低调收购了成立仅 22 个月的高科技企业 Android 及其团队。Andy Rubin 成为 Google 公司工程部副总裁，继续负责 Android 项目。

2007 年 11 月，Google 公司正式向外界展示了这款名为 Android 的操作系统，并且在当天 Google 宣布建立一个全球性的联盟组织开放手持设备联盟 (Open Handset Alliance)，该组织初始由 34 家手机制造商、软件开发商、电信运营商以及芯片制造商组成。开放手持设备联盟负责研发改良 Android 系统，联盟将支持 Google 发布的手机操作系统以及应用软件，Google 以 Apache 免费开源许可证的授权方式，发布了 Android 的源代码。

2008 年 5 月，在 Google I/O 大会上，Google 提出了 Android HAL 架构图；2008 年 8 月，Android 获得了美国联邦通信委员会 (FCC) 的批准；2008 年 9 月，Google 正式发布了 Android 1.0 系统，这也是 Android 系统最早的版本。

2010 年 2 月，Linux 内核开发者 Greg Kroah-Hartman 将 Android 的驱动程序从 Linux 内核“状态树” (“staging tree”) 上除去，从此，Android 与 Linux 开发主流分道扬镳。

2010 年 10 月，Google 宣布 Android 系统达到了第一个里程碑，即电子市场上获得官方数字认证的 Android 应用数量已经达到了 10 万个，Android 系统的应用增长非常迅速。

2011 年 1 月，Google 每日的 Android 设备新用户数量达到了 30 万部，到 2011 年 7 月，这个数字增长到 55 万部，而 Android 系统设备的用户总数达到了 1.35 亿，Android 系统已经成为智能手机领域占有量最高的系统。



2011年8月,Android手机已占据全球智能机市场48%的份额,并在亚太地区市场占据统治地位,终结了Symbian(塞班系统)的霸主地位,跃居全球第一。

2011年9月,Android系统的应用数目已经达到了48万个,而在智能手机市场,Android系统的占有率已经达到了43%,继续排在移动操作系统首位。

2012年1月,Google Android Market已有10万开发者推出超过40万活跃的应用,大多数的应用程序为免费。Android Market应用程序商店目录在新年首周的周末突破40万基准,距离突破30万应用仅用4个月。在2011年早些时候,Android Market从20万增加到30万应用也用了4个月。

Android在正式发行之之前,最开始拥有两个内部测试版本,并且以著名的机器人名称对其命名,它们分别是:阿童木(Astro),发条机器人(Bender)。后来由于涉及版权问题,Google将其命名规则变更为甜点作为它们系统版本代号的命名方法,甜点命名法开始于Android 1.5发布的时候。作为每个版本代表的甜点尺寸越变越大,然后按照26个字母数字:纸杯蛋糕(Cupcake, Android 1.5),甜甜圈(Donut, Android 1.6),松饼(Eclair, Android 2.0/2.1),冻酸奶(Froyo, Android 2.2),姜饼(Gingerbread, Android 2.3),蜂巢(Honeycomb, Android 3.0/3.1/3.2),冰淇淋三明治(Ice Cream Sandwich, Android 4.0),果冻豆(Jelly Bean, Android 4.1/4.2),奇巧巧克力(KitKat, Android 4.4),棒棒糖(Lollipop, Android 5.0),棉花糖(Marshmallow, Android 6.0)。Android版本信息见表1.1。

表 1.1 Android 版本信息

Android 版本	发布日期	代号
Android 1.0	2008年9月	Astro(铁臂阿童木)
Android 1.1	2009年2月	Bender(发条机器人)
Android 1.5	2009年4月	Cupcake(纸杯蛋糕)
Android 1.6	2009年9月	Donut(甜甜圈)
Android 2.0/2.1	2009年10月	Eclair(松饼)
Android 2.2	2010年5月	Froyo(冻酸奶)
Android 2.3	2010年12月	Gingerbread(姜饼)
Android 3.0/3.1/3.2	2011年2月	Honeycomb(蜂巢)
Android 4.0	2011年10月	Ice Cream Sandwich(冰淇淋三明治)
Android 4.1	2012年6月	Jelly Bean(果冻豆)
Android 4.2	2012年10月	Jelly Bean(果冻豆)
Android 4.4	2013年9月	KitKat(奇巧巧克力)
Android 5.0	2014年10月	Lollipop(棒棒糖)/ Android L
Android 6.0	2015年5月	Marshmallow(棉花糖)/ Android M
Android 7.0	2016年5月	/ Android N

2016 年 5 月, Google 发布了目前最新的版本 Android 7.0, 它主要的更新包括: 分屏多任务; 全新下拉快捷开关页; 通知消息快捷回复; 通知消息归拢; 夜间模式; 流量保护模式; 全新设置样式; 改进的 Doze 休眠机制; 系统级电话黑名单功能; 菜单键快速应用切换, 它对应的甜点代号尚未公布。

## 1.2 Android 平台架构及特性

Android 系统采用层次化的架构, 如图 1.1 所示, 包括 4 个功能层, 自下向上依次为: Linux 内核 (Linux Kernel)、函数库和运行时 (Libraries 和 Android Runtime)、应用程序框架 (Application Framework) 和应用程序层 (Applications)。

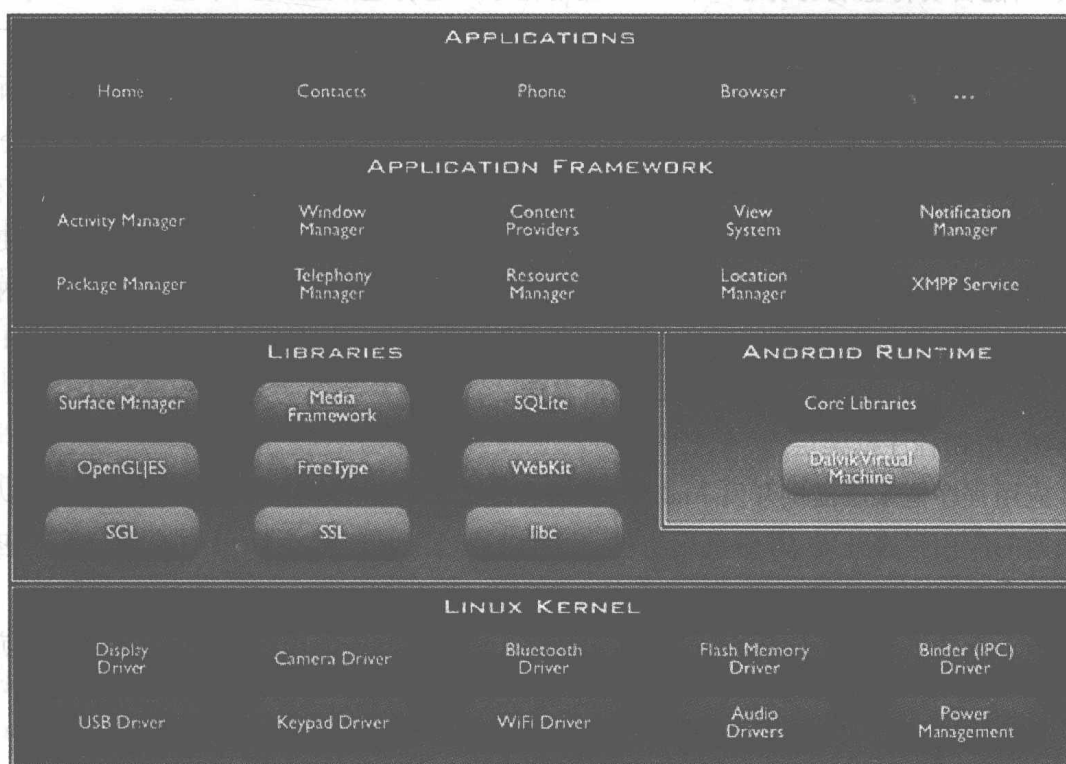


图 1.1 Android 架构

### (1) Linux 内核。

Android 系统以 Linux 操作系统内核为基础, 借助 Linux 内核服务实现硬件设备驱动、进程和内存管理、网络协议栈、电源管理以及无线通信等核心功能。自 Android 4.0 版本之后, 开始采用更新的 Linux 3.X 内核, 目前 Android 6.0 版本采用的是 Linux 3.4 内核。

与此同时, Android 内核对 Linux 内核进行了增强, 增加了一些面向移动计算的特有功能。例如: 可以根据需要杀死进程来释放需要的内存的低内存管理器 (Low Memory Keler), 为进程之间提供共享内存资源, 同时为内核提供回收和管理内存的匿名共享内

存(Ashmem)机制,以及类似于 COM 和 CORBA 分布式组件架构的轻量级的进程间通信 Binder 机制。这些内核的增强使 Android 在继承 Linux 内核安全机制的同时,进一步提升了内存管理、进程间通信等方面的安全性。

### (2) 函数库和运行时。

函数库大部分由 C/C++ 编写,所提供的功能通过 Android 应用程序框架为开发者所使用,典型的功能包括:专门为基于嵌入式 Linux 的设备定制的系统 C 库、支持多种常用音频、视频以及静态图像文件的媒体库、2D/3D 图形引擎、Web 浏览器的软件引擎、安全套接层、SQLite 数据库引擎等功能。除此之外,还有 Android NDK(Native Development Kit),即 Android 原生库,其直接使用 Android 系统资源,并采用 C 或 C++ 语言编写程序的接口,能自动将生成的动态库和 Java 应用程序一起打包成应用程序包文件,即 .apk 文件,但其安全性和兼容性可能无法保障。

Android 运行时包含:核心库和 Dalvik 虚拟机。核心库提供了大多数 Java 语言所需要调用的功能函数,并提供 Android 的核心 API,如 android.os, android.net, android.media 等。Dalvik 虚拟机是基于 Apache 的 Java 虚拟机(JVM),并被改进以此来适用于低内存、低处理器速度的移动设备环境。Dalvik 是基于寄存器,一般认为,基于寄存器的实现使用等长指令,在效率速度上较传统基于栈的 JVM 更有优势,并且 Dalvik 允许在有限的内存中同时高效地运行多个虚拟机的实例,并且每一个 Dalvik 应用作为一个独立的 Linux 进程执行,都拥有一个独立的 Dalvik 虚拟机实例。

### (3) 应用程序框架层。

应用程序框架层提供开发 Android 应用程序所需的一系列类库,使开发人员可以进行快速的应用程序开发,方便重用组件,也可以通过继承实现个性化的扩展。具体包括的模块如下:

- 活动管理器(Activity Manager):管理应用程序的生命周期,并提供常用的导航回退功能,为所有程序的窗口提供交互的接口;
- 窗口管理器(Window Manager):对所有开启的窗口程序进行管理;
- 内容提供者(Content Provider):提供一个应用程序访问另一个应用程序数据(例如:联系人数据库)的功能,或者实现应用程序之间的数据共享;
- 视图系统(View System):构建应用程序的基本组件,包括列表(lists),网格(grid),文本框(text boxes),按钮(buttons),还有可嵌入的 web 浏览器;
- 通知管理器(Notification Manager):使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):对应用程序进行管理,提供的功能,例如:安装应用程序、卸载应用程序、查询相关权限信息等;
- 资源管理器(Resource Manager):提供对非代码资源的访问,例如:本地化字符

串、图片、音频、布局文件等；

- 位置管理器 (Location Manager) : 提供位置服务；
- 电话管理器 (Telephony Manager) : 管理所有的移动设备功能；
- XMPP 服务 : 是 Google 在线即时交流软件中一个通用的进程, 提供后台推送服务。

(4) 应用层。

Android 平台的应用层上包括各类与用户直接交互的应用程序, 或由 Java 语言编写的在后台运行的服务程序。例如, 与 Android 系统一起发布的核心应用程序, 典型的包括: Email 客户端、SMS 短消息程序、电话拨号及联系人管理程序、浏览器、日历、地图等程序, 以及第三方开发人员开发的其他应用程序。

### 1.3 开发环境的搭建

目前, 主流的 Android 开发环境有两类: 一类是采用 Eclipse + ADT + Android SDK 的组合方式, 另一类是 Android Studio + Android SDK 的组合方式。前者是国内最为普及的开发环境, 学校以及软件公司的主流选择, 目前 Google 停止了对 ADT 插件的更新, 因此, 后者将逐渐取代前者成为今后的唯一选择。因此, 本书对这两种开发环境的搭建过程均进行了详细的介绍, 但是考虑到学生在实际学习过程中对 Eclipse 更为熟悉, 以及未来就业之后的便捷, 本书后续章节的示例均在 Eclipse + ADT + Android SDK 开发环境下予以介绍。

#### 1.3.1 Eclipse + ADT

本书讲解的 Eclipse + ADT + Android SDK 开发环境安装是以 Window 10 为平台, 安装的软件为 JDK 1.8、Eclipse Mars.2 (4.5.2) Release for Windows (64bit)、ADT 23.0.7 和 Android SDK r24.3.4。

(1) 安装 JDK。

安装 Eclipse 集成开发环境之前, 首先需要安装 JRE, 在 Window 10 上安装 JDK 1.8 (包含 JRE) 的方法如下:

① 打开下载网址: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>, 如图 1.2 所示, 根据自己所拥有的计算机配置选择下载软件版本, 此处本书选择的 Windows x64, 即 jdk-8u77-windows-x64.exe。

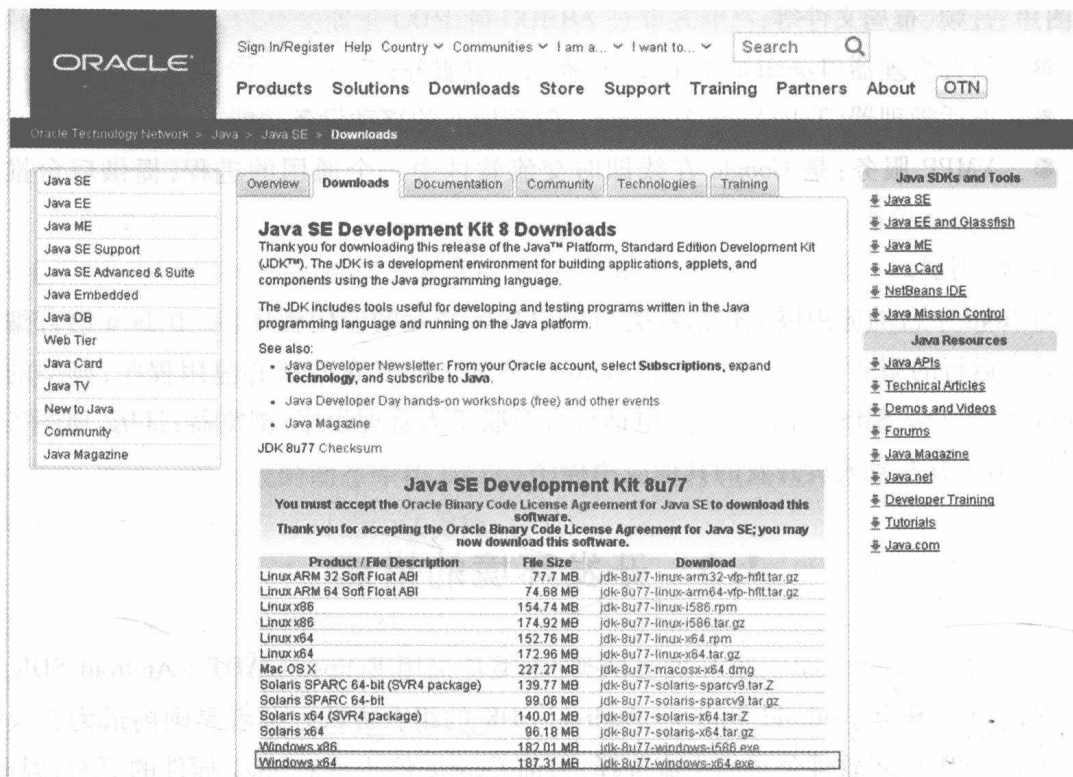


图 1.2 下载页面

②下载之后，双击文件“jdk-8u77-windows-x64.exe”，打开安装程序，依次选择默认选项。

安装完成之后，需要确认一下是否安装成功，在 Windows 10 平台中，通过同时按下“win + R”，在打开的对话框中输入“cmd”并按下“确定”，在打开的 CMD 窗口中输入“java-version”命令，如果显示如图 1.3 所示的提示信息，说明安装成功。

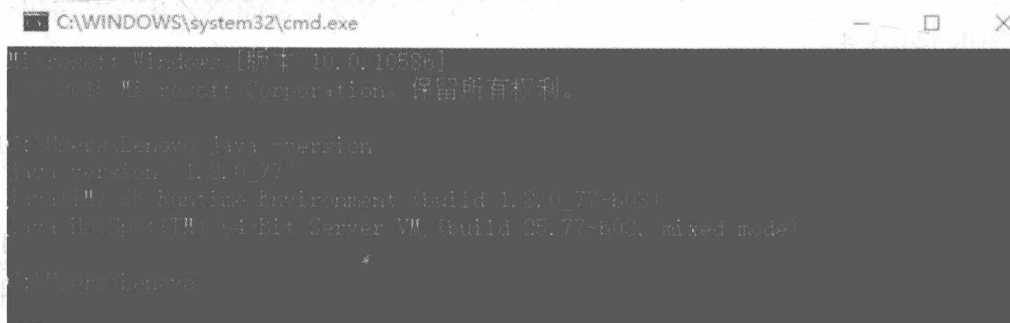


图 1.3 CMD 窗口

如果经过上述步骤发现安装失败，需要进行路径配置。在桌面上右击“此电脑(或计算机)”图标，在弹出的快捷菜单中选择“属性”命令，在弹出的对话框中选择“高级系统



设置”选项卡,在弹出的对话框中选择“环境变量”按钮,弹出“环境变量”对话框,如图 1.4 所示。

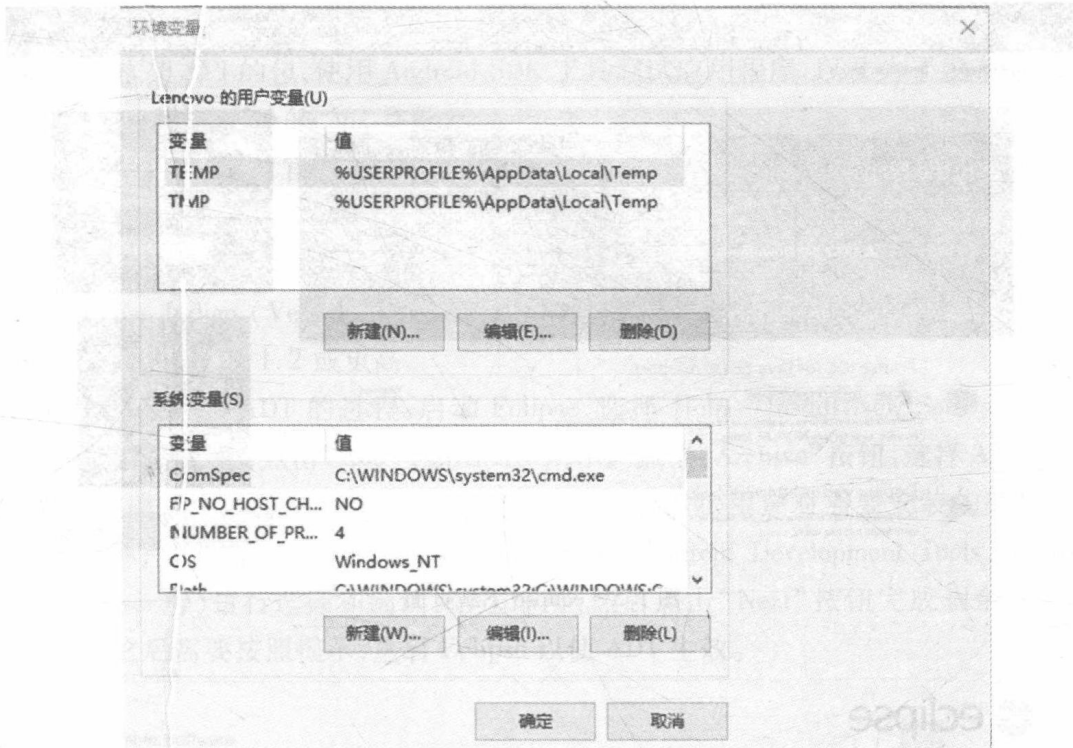


图 1.4 “环境变量”对话框

接下来,设置环境变量的步骤如下:

- “新建”一个“系统变量”,在“变量名”中输入“JAVA\_HOME”,在“变量值”中输入 JDK1.8 的安装目录,本书为“C:\Program Files\Java\jdk1.8.0\_77”;
- “新建”一个“系统变量”,在“变量名”中输入“classpath”,在“变量值”中输入:.;%JAVA\_HOME%\lib\rt.jar;%JAVA\_HOME%\lib\tools.jar;
- “编辑”一个“变量名”为“Path”的系统变量,在“变量值”最前面添加:%JAVA\_HOME%\bin。

## (2) 安装 Eclipse。

①打开 Eclipse 的官方下载网址:<http://www.eclipse.org/downloads/>,如图 1.5 所示,此处选择 Eclipse Installer for Windows 64bit。

②来到镜像页面,如图 1.6 所示,选择最近的镜像,此处推荐的下载镜像为“China-University of Science and Technology of China (http)”。

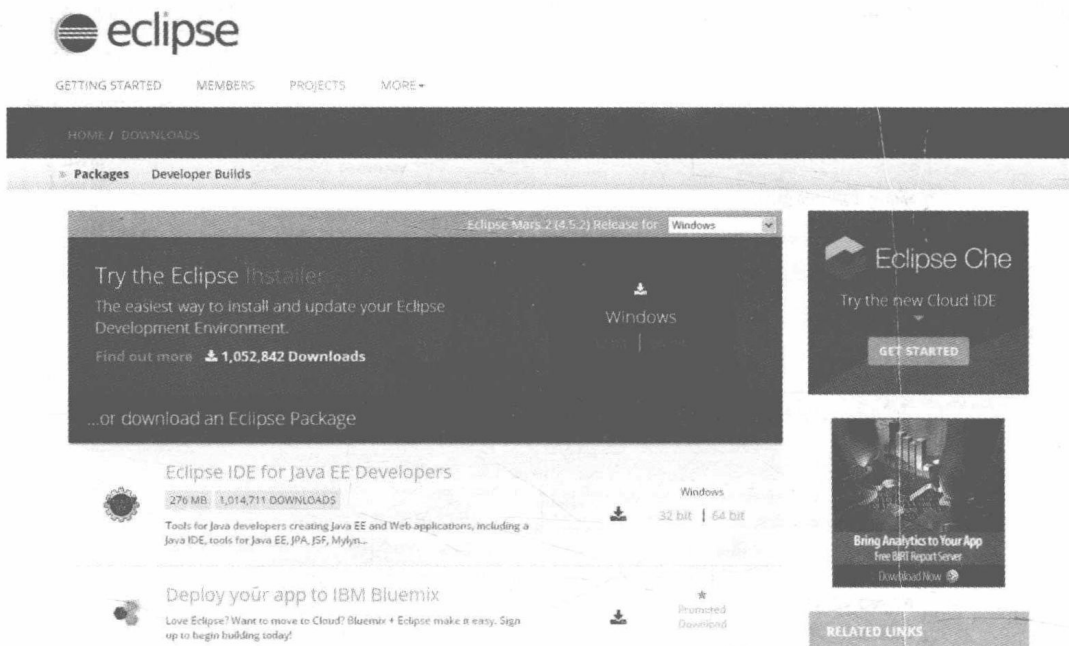


图 1.5 eclipse 下载页面

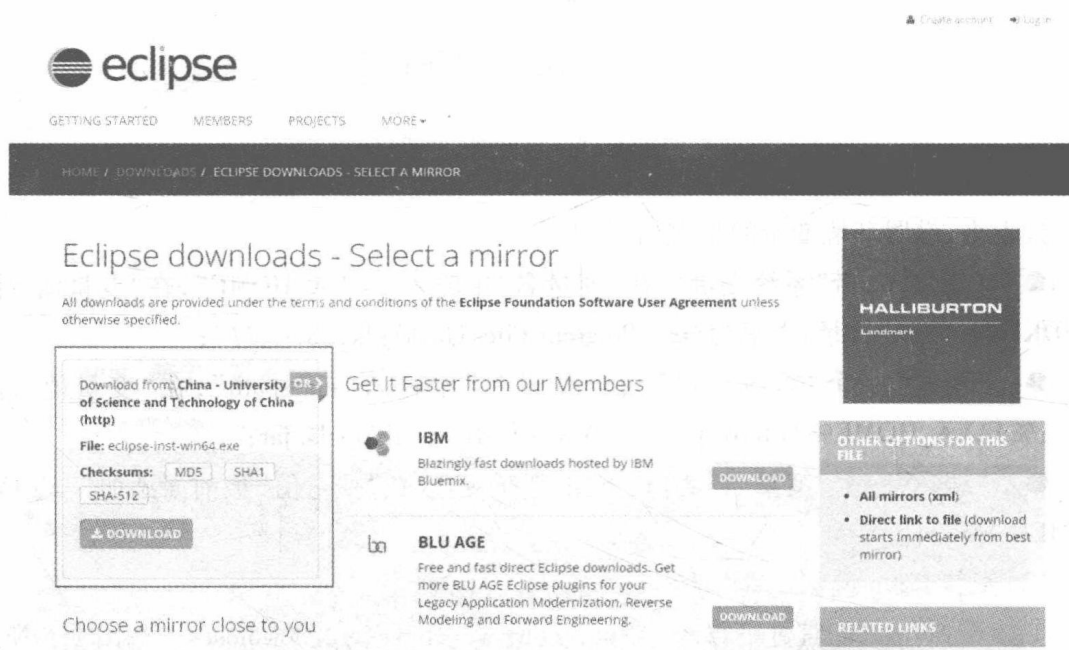


图 1.6 选择镜像

③下载完成之后,双击文件“eclipse-inst-win64.exe”,打开安装程序,选择版本类型 Eclipse IDE for Java Developers,然后依次选择默认选项即可,过程不再赘述。本书将 Eclipse 安装到目录:E:\Eclipse。打开目录,双击“eclipse.exe”图标,Eclipse 能自动找到先前安装的 JDK 路径。

### (3) 安装 ADT。

Android 开发工具(Android Development Tools, ADT)是一款支持 Eclipse IDE 的插件,它扩展了 Eclipse 的能力,使其能够快速地建立新的 Android 工程,创建应用程序 UI,添加基于 Android 框架 API 的包,使用 Android SDK 工具调试应用程序,以及为了发布应用程序而导出签名(或未签名)的 .apk 文件。

2015 年 8 月,Android 官方网站发布了最新版本 ADT 23.0.7(目前 ADT 的更新已经结束),下载地址为“<http://pan.baidu.com/s/1boH9obP>”,该版本的使用约束包括:

- Java 7 或更高;
- Eclipse Indigo (Version 3.7.2) 或更高;
- SDK Tools r24.1.2 或更高。

在 Eclipse 中安装 ADT 的过程:启动 Eclipse,选择 Help→Install New Software,打开 Eclipse 的插件安装界面,点击“Add”,如图 1.7 所示。点击“Archive”按钮,选择 ADT 插件压缩包在本地磁盘中的位置。在 ADT 插件安装前,会提示用户对需要安装的插件(即 Developer Tools)的具体内容(例如:Android DDMS、Android Development Tools、Android Hierarchy Viewer 等)进行选择 and 确认,全选即可,然后点击“Next”按钮完成剩余安装步骤,安装成功之后需要按照提示,重启 Eclipse 以使 ADT 生效。

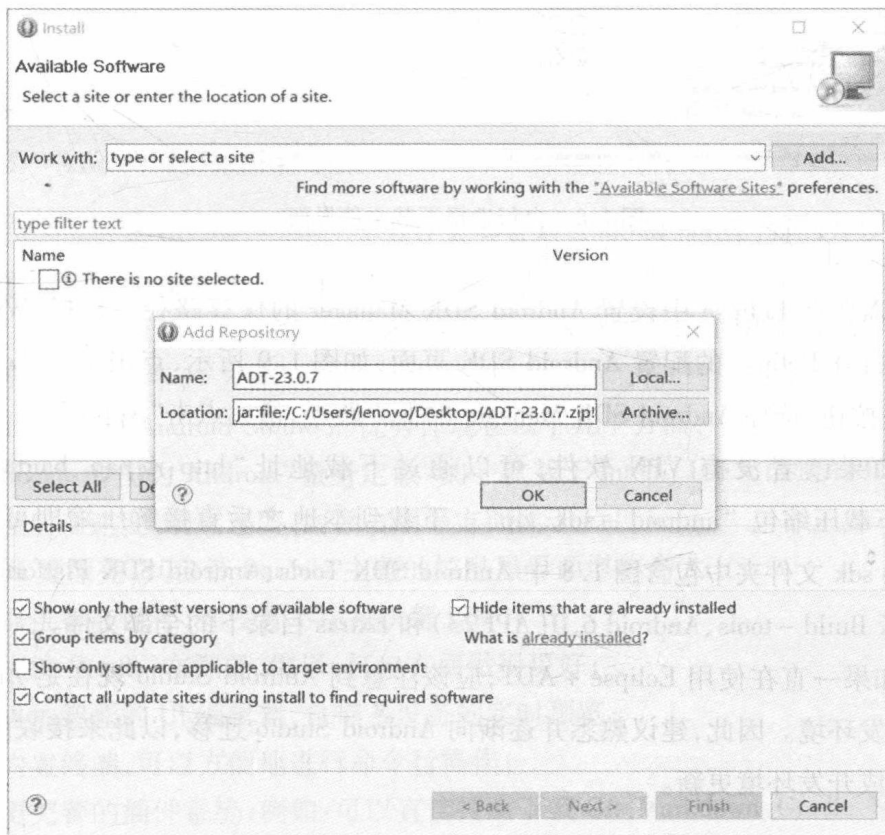


图 1.7 本地手动安装 ADT

**注意:**由于在大陆地区无法访问 Android 官方网站,因此,在“Location”中添加网址“<http://dl-sll.google.com/Android/eclipse/>”,进行在线安装时,需要借助 VPN 软件。

#### (4) 安装 Android SDK Manager。

下载并安装 Android SDK Manager,本书提供的版本是 installer\_r24.3.4-windows.exe (2015年8月更新),其下载地址为“<http://pan.baidu.com/s/1nvkvfBz>”。安装成功之后,如图 1.8 所示,可以自行选择需要下载的文件,点击右下角按钮开始下载/安装(此时需要借助 VPN 软件)。

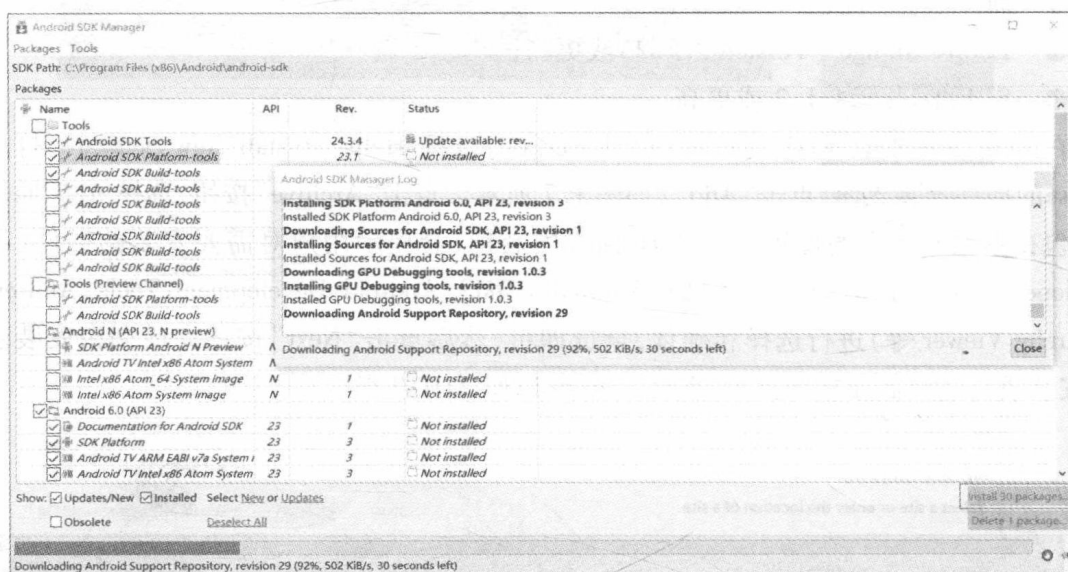


图 1.8 自行选择下载文件界面

下一步需要在 Eclipse 中设置 Android SDK Manager 的保存路径。选择:Windows→Preferences,打开 Eclipse 的配置 Android SDK 页面,如图 1.9 所示,点击 SDK Location 后面的 Browse 按钮,选择 Android SDK Manager 的保存路径,最后点击“Apply”。

**注意:**如果读者没有 VPN 软件,可以通过下载地址“<http://pan.baidu.com/s/1i45RH1j>”下载压缩包“android-sdk.zip”。下载到本地之后直接解压缩即可,无须安装,android-sdk 文件夹中包含图 1.8 中 Android SDK Tools、Android SDK Platform-tools、Android SDK Build-tools、Android 6.0(API 23)和 Extras 目录下的全部文件。

**注意:**如果一直在使用 Eclipse + ADT,应该注意到 Android Studio 现在是 Android 官方的集成开发环境。因此,建议熟悉并逐渐向 Android Studio 迁移,以此来接收所有的最新版本的集成开发环境更新。