



Android

开发案例教程与项目实战

在线实验 + 在线测试

西普教育研究院 林雪纲 时允田 主编

薛乔毓 肖何 马云涛 副主编

- ◆ 内容新颖，可操作性强，层层深入，简明易懂。
- ◆ 从实用角度出发，重点培养动手解决问题的能力。
- ◆ 提供体系完整的 100 学时在线实验，即学即练，书网结合。
- ◆ 112 个案例实战（课程配套案例 52 个、扩展案例 60 个）和 50 余套自测题，附赠 PPT 课件、案例源代码、视频等资源。



让实验更简单



开放实验云平台



Android

开发案例教程与项目实战

在线实验 + 在线测试

西普教育研究院 林雪纲 时允田 主编

薛乔毓 肖何 马云涛 副主编

人民邮电出版社

北京

图书在版编目 (CIP) 数据

Android开发案例教程与项目实战：在线实验+在线自测 / 林雪纲, 时允田主编. -- 北京：人民邮电出版社, 2016.11

ISBN 978-7-115-43767-9

I. ①A… II. ①林… ②时… 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243358号

内 容 提 要

本书分基础篇和提高篇两部分，共11章。基础篇包括第1~5章，提高篇包括第6~11章。第1章是初识Android，介绍Android基本知识、环境的安装与搭建。第2章是Android核心组件，介绍常用组件的功能和用法。第3章是界面视图，介绍常用视图组件的使用。第4章是数据存储，介绍常用的数据存储方式。第5章是项目实战：实验吧，串联前4章所学知识进行综合应用。第6章是多媒体，介绍MediaPlayer和动画的应用。第7章是数据解析，介绍XML和JSON数据的解析方式。第8章是网络编程，介绍从网络中获取资源和上传信息的方法。第9章是图形绘制，介绍绘制2D图形和3D图形的方法。第10章是传感器，介绍传感器的分类及加速度传感器的应用。第11章是项目实战：新闻速览，该案例回顾全书的内容，促进读者对所学的知识进行总结和巩固。全书结构合理，知识点全面，讲解详细，重点、难点突出。

本书适合作为Android移动开发的入门参考书，也可作为院校计算机及相关专业的教材。

-
- ◆ 主 编 西普教育研究院 林雪纲 时允田
 - 副 主 编 薛乔毓 肖 何 马云涛
 - 责任编辑 桑 珊
 - 执行编辑 左仲海
 - 责任印制 焦志炜

 - ◆ 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>
 - 固安县铭成印刷有限公司印刷

 - ◆ 开本：787×1092 1/16
 - 印张：19.5 2016年11月第1版
 - 字数：477千字 2016年11月河北第1次印刷
-

定价：49.80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

FOREWORD

编写目的

Android 系统的崛起和塞班系统的衰退,标志着智能移动时代的到来。各大移动终端厂商和移动系统企业之间的竞争甚是激烈,Android 系统的手机也受到越来越多使用者的欢迎。现如今,Android 移动开发工作者越来越多,相应课程也成为高校计算机相关专业必修的课程之一。

平台支撑

为了让广大学习者能够快速入门,本书以实践案例为主线,通过遵循书中的案例操作步骤,完成一个个实验案例,学习 Android 开发技术。同时,北京西普阳光教育科技股份有限公司(简称西普教育)开发的在线教育平台——实验吧(<http://www.shiyanbar.com>),提供了强大的集成实验环境及海量的在线教学资源,把本书配套的实验搬到线上,可以让读者更方便地结合本书进行动手实践。

1. 如何学习本书中配套实验课程

(1) 购买本书后,找到粘贴在本书封底的刮刮卡,刮开获得学号。

(2) 登录实验吧网站(www.shiyanbar.com),完成网站注册。

(3) 登录人邮学院在线实验中心(rymooc.shiyanbar.com),输入在实验吧注册的账户及密码,完成登录(见图 1)。

(4) 输入刮刮卡中的学号,姓名填写“人邮学院”,点击保存,完成绑定(见图 2)。



图 1 登录在线实验平台

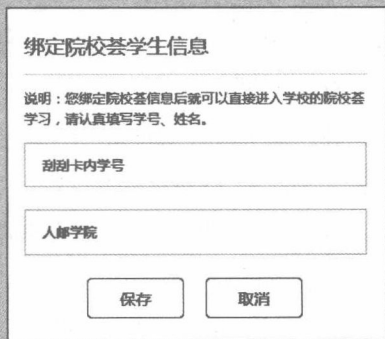


图 2 绑定学生信息

(5) 完成绑定后，自动登录进入在线实验中心，开始学习本书配套课程资源。

2. 如何学习本书中配套练习题

实验吧教研团队为本书配套了丰富的课后练习题，读者通过扫描本书章节里配套的习题二维码，即可进行在线自测，提交后自动判断正误，并提供正确答案（见图3）。

本书特点

本书编者有着多年的 Android 实际项目开发经验与丰富的高校教学和 IT 培训经验。编者带领西普教育研究院的 Android 师资队伍在南京大学、大连理工大学等高校专业课程教学和学生培训过程中进行了多次基于案例化教学、在线实验、翻转课堂的创新教学实践，本书也是系列教学实践的成果之一。

本书主要特点如下。

1. 真正以实际案例为驱动

为了使读者能快速地掌握相关技术，本书各章都以实际案例来驱动学习，让读者真正地实现边实操边学习。

2. 合理、高效的组织结构

本书采用“做、学、教一体化”的教学方法，培养读者的主观能动性，在教与学的过程中弱化“教”，深化“做”与“学”，最终达到“老师轻松教，学生高效学，实验简单做”的目的。

3. 内容线下线上同步，丰富实用

本书的训练紧紧围绕着与实际相结合的实验进行，各章做简要的知识点分析后，读者可以随时随地在线进行对应实验，免去了读者配置开发环境的困扰。

资源下载

为方便读者使用，本书配有全部实例的源代码及电子教案。读者可通过本书配送的账号登录 <http://rymooc.shiyanbar.com/> 网站进行资料下载与在线学习。

致谢

本书由林雪纲、时允田任主编，薛乔毓、肖何、马云涛任副主编。编写过程得到西普教育研究院和实验吧团队的大力协助，特别感谢吴新国、张慧、王利涛、张美丽等全力支持。由于编者水平有限，书中难免存在不妥之处，敬请广大读者批评指正。

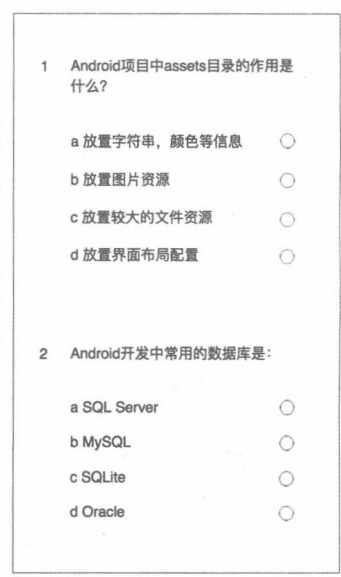


图3 在线测试

编者

2016年8月

目 录 CONTENTS

基础篇

第 1 章 初识 Android	1	3.1.1 View	47
1.1 Android 简介	1	3.1.2 ViewGroup	47
1.2 Android 平台的特性	1	3.1.3 布局参数	48
1.3 Android 的体系架构	2	3.2 界面布局	49
1.4 Android 开发环境搭建	4	3.2.1 线性布局	49
1.4.1 JDK 安装过程	4	3.2.2 相对布局	52
1.4.2 配置 JDK 环境变量	6	3.2.3 帧布局	55
1.4.3 Eclipse 的安装与启动	7	3.2.4 表格布局	56
1.4.4 安装 ADT 插件	8	3.3 界面组件视图	59
1.4.5 配置 Android SDK	10	3.3.1 文本视图 (TextView)	60
1.4.6 创建 AVD	12	3.3.2 编辑框 (EditText)	63
1.5 创建 Android 应用程序	14	3.3.3 图片控件 (ImageView)	65
1.6 Android 程序目录结构	17	3.3.4 自动提示控件 (AutoCompleteTextView)	68
1.7 学习 Android 必备知识	18	3.3.5 按钮 (Button)	70
本章小结	18	3.3.6 下拉菜单 (Spinner)	72
习题	19	3.3.7 复选按钮 (CheckBox)	75
第 2 章 Android 核心组件	20	3.3.8 单选按钮 (RadioButton)	78
2.1 Activity	20	3.3.9 进度条 (ProgressBar)	81
2.2 Intent	27	3.3.10 拖动条 (SeekBar)	83
2.3 Service	31	3.3.11 星级评分条 (RatingBar)	85
2.3.1 Service 的生命周期	32	3.3.12 单击事件处理	86
2.3.2 Service 的两种启动方式	33	3.3.13 视图列表 (ListView)	88
2.4 ContentProvider	38	3.3.14 网格视图 (GridView)	92
2.5 BroadcastReceiver	43	3.3.15 适配器 (Adapter)	94
本章小结	46	3.3.16 画廊 (ImageSwitcher、 Gallery)	99
习题	46	3.3.17 ViewPager	102
第 3 章 界面视图	47		
3.1 布局管理视图	47		

Android 开发案例教程与项目实战 (在线实验+在线自测)

3.3.18	Fragment	111	第 5 章 项目实战: 实验吧	139
3.3.19	Handler	119	5.1 项目概述	139
本章小结		121	5.2 功能需求	139
习题		122	5.3 软件开发需求	140
第 4 章 数据存储		123	5.4 效果展示	140
4.1	SharedPreferences 存储	123	5.5 系统开发	141
4.1.1	操作模式	123	5.6 打包与运行	162
4.1.2	数据存取	124	5.6.1 Android Tools 向导	163
4.2	File 存储	128	5.6.2 选择项目名称	163
4.2.1	内部存储	129	5.6.3 选择或创建数字证书	164
4.2.2	外部存储	130	5.6.4 指定 APK 文件存储	
4.3	SQLite	134	目录	164
本章小结		138	本章小结	165
习题		138	习题	165

提高篇

第 6 章 多媒体	166	7.2.2 JSON 的结构	203	
6.1	MediaPlayer	166	7.2.3 JSON 和 XML	204
6.1.1	通过静态方法 create() 装载 资源实现播放	166	本章小结	211
6.1.2	通过 setDataSource() 方法 装载音频文件实现播放	167	习题	211
6.2	Animations 动画	172	第 8 章 网络编程	212
6.2.1	AlphaAnimation	172	8.1 网络基础	212
6.2.2	ScaleAnimation	174	8.1.1 TCP/IP 协议	212
6.2.3	RotateAnimation	177	8.1.2 UDP 协议	213
6.2.4	TranslateAnimation	180	8.2 HTTP 应用	213
6.2.5	FrameAnimation	182	8.2.1 HTTP 应用——java.net 实现	213
本章小结		186	8.2.2 HTTP 应用——通过 HttpClient 实现	214
习题		186	8.2.3 HttpClient 操作步骤	214
第 7 章 数据解析	187	8.3 HttpURLConnection	221	
7.1	XML 解析	187	8.4 WebService	229
7.1.1	DOM 解析器	187	8.5 WiFi	240
7.1.2	SAX 解析器	192	本章小结	248
7.1.3	PULL 解析器	197	习题	248
7.2	JSON 解析	202	第 9 章 图形绘制	249
7.2.1	JSON 定义	202	9.1 2D 图形绘制	249

9.1.1 绘图类	249	10.11 线性加速度传感器	263
9.1.2 常见绘图类型	250	10.12 旋转矢量传感器	264
9.2 OpenGL	254	本章小结	271
9.2.1 OpenGL 简介	255	习题	271
9.2.2 Android 支持 OpenGL 列表	255	第 11 章 项目实战:	
9.2.3 OpenGL API	255	新闻速览	272
本章小结	259	11.1 项目概述	272
习题	259	11.2 功能需求	272
第 10 章 传感器	260	11.3 软件开发需求	273
10.1 传感器	260	11.4 效果展示	273
10.2 加速度传感器	261	11.5 系统开发	273
10.3 磁力传感器	262	11.6 打包与运行	301
10.4 方向传感器	262	11.6.1 Android Tools 向导	302
10.5 陀螺仪传感器	262	11.6.2 选择项目名称	302
10.6 光线感应传感器	263	11.6.3 选择或创建数字证书	303
10.7 压力传感器	263	11.6.4 指定 APK 文件存储	
10.8 温度传感器	263	目录	303
10.9 接近传感器	263	本章小结	304
10.10 重力传感器	263	习题	304

基础篇



第 1 章 初识 Android



本章要点

- Android 简介
- Android 体系架构
- Android 环境搭建
- 创建 Android 应用程序



引言

随着移动智能设备的普及和发展,移动软件的开发越发受到开发者的青睐。在三大移动开发领域(iOS、Android、Windows Phone)中以 Android 发展最为迅猛,在国内自 2009 年开始,短短几年时间,就在移动开发领域占据一席之地。本章通过对 Android 的体系架构、Android 开发环境的搭建、Android 项目的开发步骤以及 Android 第一个应用项目的介绍,让读者对 Android 有一个初步的认识。

1.1 Android 简介

Android 一词的英文本意是指“机器人”,是 Google 公司专门为移动设备开发的移动应用开发平台,其中包含操作系统、中间件和核心应用等。Android 公司最早由安迪·鲁宾创办并于 2005 年被搜索巨人 Google 收购。2007 年 11 月 5 日,Google 正式发布 Android 平台。在 2010 年年底,Android 超越称霸 10 年的诺基亚 Symbian 系统,成为全球最受欢迎的智能手机平台之一。

概括地说,Android 是基于 Java 并运行在 Linux 内核上的轻量级的、功能全面的操作系统,在这个系统上 Google 公司内置很多应用,如打电话、发短信、个人信息管理、浏览互联网等。

1.2 Android 平台的特性

1. 开放性

Android 平台具有开放性。首先是 Android 源码的开放,每一个应用程序可以调用其内部的核心应用源码;其次是平台的开放,Android 平台不存在任何阻碍移动产业创新的

Android 开发案例教程与项目实战（在线实验+在线自测）

专有限制，任何联盟厂商都可以根据自己的需要自行定制基于 Android 操作系统的手机产品；再次是运营的开放，手机使用什么方式接入网络，已不再受运营商的控制，用户可以更加方便地连接网络等。

2. 不受运营商的束缚

在 Android 平台中，其内部的核心应用和第三方的应用是完全平等的，用户能完全根据自己的喜好使用它们来定制手机服务系统；其应用程序框架支持组件的重用与替换，开发者可以完全平等地调用其内部核心程序或第三方应用程序。

3. 丰富的硬件选择

Android 平台支持丰富的硬件，这一点还是与 Android 平台的开放性相关，由于 Android 的开放性，众多的厂商会推出功能特色各异的多产品。

4. 不受任何限制的开发商

Android 平台提供给第三方开发商一个十分宽泛、自由的环境，因此不会受到各种条件的限制，这样才会产生各种别样的应用软件。当然也会有一些不健康的、恶意的程序和游戏出现，抑制这样的软件，需要 Android 平台不断去优化和完善。

5. 无缝结合的 Google 应用

Google 服务如地图、邮件、搜索等已经成为连接用户和互联网的重要纽带，Android 平台恰恰能够将移动设备与这些优秀的服务软件无缝接合。

1.3 Android 的体系架构

Android 平台具有一个完善的体系架构，图 1-1 展示了 Android 操作系统的主要组成部分。



图 1-1 Android 的系统架构

下面通过对每一个部分的讲解,让读者对 Android 系统有一个初步的认识。如图 1-1 所示,Android 系统架构分为三层五部分。

1. Applications(应用层)

Android 中的应用程序是用 Java 语言编写的运行在虚拟机上的程序。Android 系统中自带一些应用程序,包括电子邮件客户端、SMS 程序、日历、地图、浏览器、联系人等。

2. Application Framework(应用框架层)

这一层是 Google 发布的核心 API 框架。通过提供一个自由开放的平台,为开发者开发出丰富而又创新的应用提供了良好的支持。开发者可以自由地利用现有的硬件设备、信息访问位置的优势、后台的运行服务、设置报警器、添加通知到状态条等。

开发者可以访问 Android 提供的 API,并且它的设计很好地实现了组件的重用。任何一个应用都可以公布出来,其他的应用也可以使用这些功能,并且该机制也是允许用户去替换使用的。所有的应用都是有一套服务和系统为基础的,主要包括以下内容。

(1) View System: 可以用来构建应用程序、它包括列表、网格、文本框、按钮以及可嵌入的 Web 浏览器。

(2) Content Providers: 它可以让一个应用访问另一个应用的数据,或共享它们自己的数据。

(3) Resource Manager: 提供非代码资源的访问,如本地字符串、图形和布局文件。

(4) Notification Manager: 应用可以在状态栏中显示自定义的提示信息。

(5) Activity Manager: 用来管理应用程序生命周期并提供常用的导航退回功能。

(6) Window Manager: 管理所有的窗口程序。

(7) Package Manager: Android 系统内的程序管理。

3. Libraries (功能组件库)

功能组件库主要提供 Android 程序运行时需要的一些类库,这些类库是使用 C/C++ 语言编写的,主要包括以下内容。

(1) 当使用 Android 应用框架时,Android 系统会通过一些 C/C++ 库支持使用各个组件。

(2) Bionic 系统 C 库: C 语言标准库,系统底层的库,通过 Linux 系统来调用。

(3) 多媒体库(Media Framework): Android 系统多媒体库,基于 Packer Video Open Core,该库支持多种常见格式的音频、视频的回放和录制以及图片,如 MPEG4、MP3、AAC、AMR、JPG、PNG 等。

(4) SGL: 2D 图形引擎库。

(5) SSL: 位于 TCP/IP 协议与各种应用层协议之间,为数据通信提供支持。

(6) OpenGL ES 1.0: 3D 效果的支持。

(7) SQLite: 关系数据库。

(8) WebKit: Web 浏览器引擎。

(9) FreeType: 位图及矢量。

4. Dalvik Virtual Machine (虚拟机)

在 Android 操作系统中,每个 Java 程序都运行在 Dalvik 虚拟机上,只能执行以.DEX 为后缀的可执行文件。当 Java 程序通过编译后,最后还需要通过 SDK 中的 dx 工具转换

成.DEX 格式才能正常在虚拟机上执行，Java 虚拟机运行的是 Java 字节码，而 Dalvik 虚拟机运行的则是其专有的文件格式.DEX 的文件。

Dalvik 虚拟机有如下几个主要特征。

(1) 专有的.DEX 文件格式。

(2) DEX 的优化，DEX 文件的结构是紧凑的。

(3) 基于寄存器。

(4) 一个应用，一个虚拟机实例，一个进程。每一个 Android 应用运行在一个 Dalvik 虚拟机实例中，每一个虚拟机实例都是一个独立的进程空间。



学习 小贴士

Android 中的 Dalvik 虚拟机和 Java 中的虚拟机有所不同，Dalvik 是基于寄存器的，而 JVM 是基于栈内存的，基于寄存器的虚拟机在编译的时候花费的时间自然就短一些。同时，Java 虚拟机运行的是 Java 字节码，而 Dalvik 虚拟机运行的是专有格式 DEX，该格式的文件比 class 文件更高效。

5. Linux Kernel (Linux 内核层)

Android 的核心系统服务是基于 Linux 2.6 内核的，如安全性、内存管理、进程管理、网络协议和驱动模型等都依赖于该内核。

1.4 Android 开发环境搭建

Android 应用程序开发，需要搭建合适的开发环境。表 1-1 列出了进行 Android 开发所必需的系统开发环境需求。同时，进行 Android 开发还需要一些合适软件的支持，主要包括 JDK 的安装、Eclipse 的安装、ADT 的安装、Android SDK 安装及创建 AVD。

表 1-1 开发环境需求

操作系统	要 求 系统版本	内 存
Windows	Windows 8/7/Vista/2003(32 或 64 位)	最小 2 GB，推荐 4 GB
Mac OS	Mac OS X 10.8.5 或更高	最小 2 GB，推荐 4 GB
Linux	Linux GNOME 或 KDE (K 桌面环境)	最小 2 GB，推荐 4 GB

1.4.1 JDK 安装过程

JDK 是 Sun 公司的产品，后来 Sun 公司被 Oracle 收购，现在下载 JDK 需要到 Oracle 公司的官方网站 (<http://www.oracle.com/index.html>) 下载。本书在进行 Android 应用开发时，使用的是 JDK7 版本，具体安装步骤如下。

(1) 由于本书使用的是 64 位的 Windows 7 操作系统，所以下载的是名称为 jdk-7u75-windows-x64.exe 的文件。双击该文件，弹出的界面如图 1-2 所示。

(2) 单击“下一步”按钮，将弹出“定制安装”对话框，在该对话框中，可以选择安装的功能组件。这里选择默认设置，如图 1-3 所示。

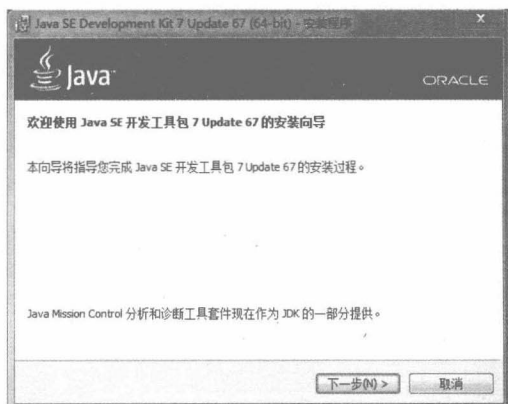


图 1-2 JDK 安装

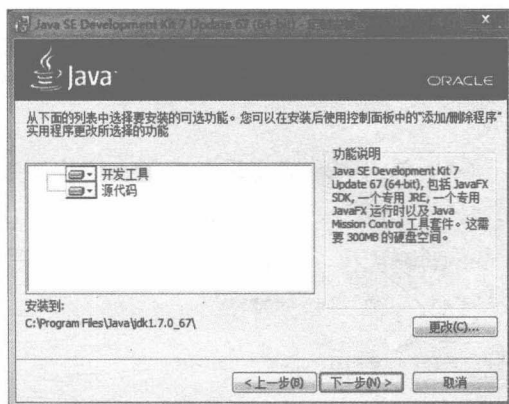


图 1-3 定制安装 (自定义安装)

(3) 可以根据需要更改安装路径, 如无特殊要求, 可以单击“下一步”按钮, 稍后会弹出 jre7 安装对话框, 如图 1-4 所示。

(4) 选择 jre7 安装目录, 如无特殊需要默认安装路径, 单击“下一步”按钮, 进入安装界面, 安装完成后会出现对话框, 单击“完成”按钮, 如图 1-5 所示。

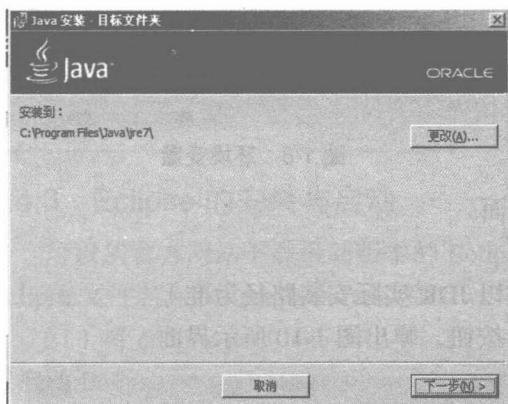


图 1-4 jre 安装对话框

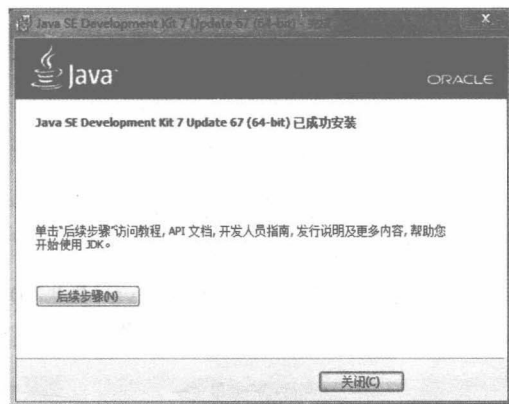


图 1-5 JDK 安装完成

(5) JDK 安装在 C:\Program Files\Java 目录下, 如图 1-6 所示。

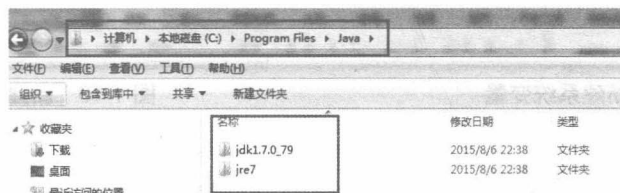


图 1-6 JDK 安装目录



学习
小贴士

在安装 JDK 时, 如果系统是 Win32 的, 那么就下载 JDK 32 位版本, 例如: 下载 jdk-7u79-window-i586.exe; 如果是 Win64 的, 那么就下载 JDK64 位版本, 例如: 下载 jdk-7u79-windows-x64.exe。

1.4.2 配置 JDK 环境变量

(1) 鼠标右键单击“计算机”→“属性”→“高级系统设置”命令，弹出图 1-7 所示界面。

(2) 单击“环境变量”按钮，弹出图 1-8 所示界面。

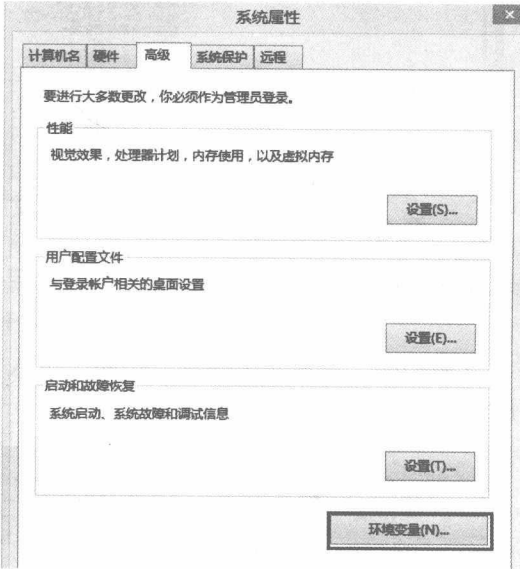


图 1-7 系统属性

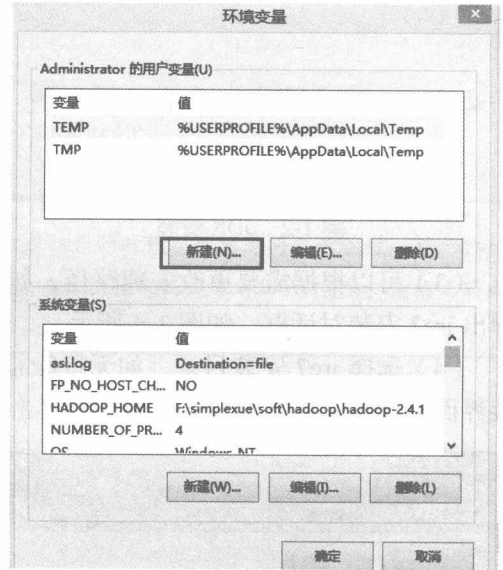


图 1-8 环境变量

(3) 单击“新建”按钮，弹出图 1-9 所示界面。

变量名: JAVA_HOME。

变量值: C:\Program Files\Java\jdk1.7.0_07 (以 JDK 实际安装路径为准)。

(4) 单击“确定”按钮，再次单击“新建”按钮，弹出图 1-10 所示界面。

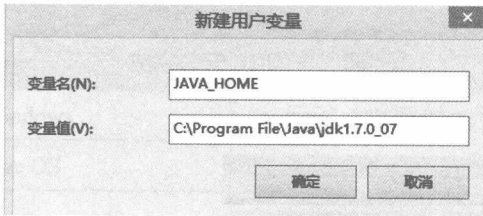


图 1-9 新建系统变量

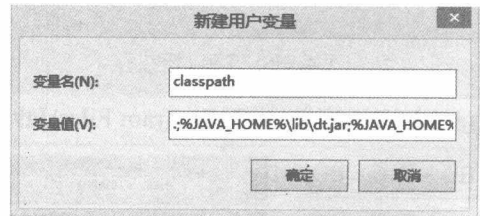


图 1-10 新建系统变量

变量名: classpath。

变量值: .;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar。

(5) 如果有 path 变量，则双击将其打开进行编辑，没有就单击“新建”按钮，如图 1-11 所示。

变量名: path。

变量值: %JAVA_HOME%\bin;D:; (注意变量值之间用“;”隔开)。

(6) 单击“开始”→“运行”→输入“javac”按“回车”键，出现命令列表，JDK 安装成功，如图 1-12 所示。

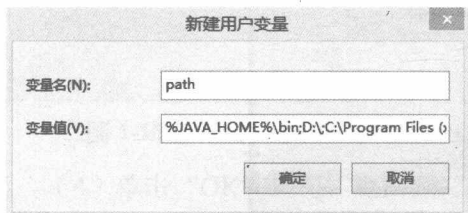


图 1-11 编辑系统变量

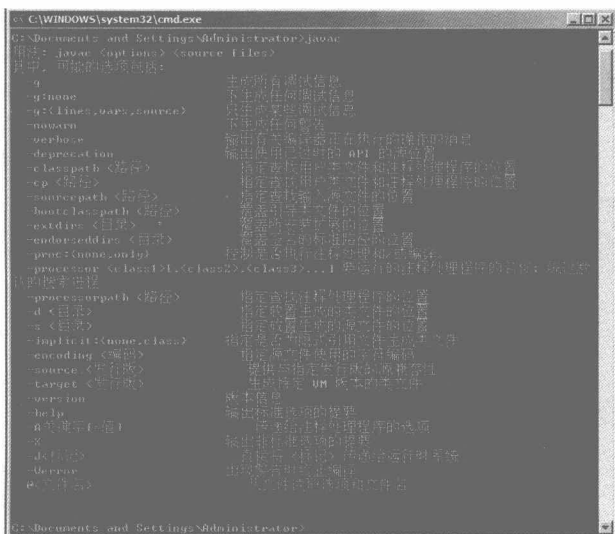


图 1-12 输入“javac”命令



学习 小贴士

安装 JDK 完毕之后，需要进行环境变量配置，注意区分用户变量和系统变量中配置 classpath 和 path 的区别，前者配置效果只对当前用户有效，后者配置对系统所有用户有效。

1.4.3 Eclipse 的安装与启动

可以从官方网站下载最新版本的 Eclipse，具体网址为 <http://www.eclipse.org>。下载后的 Eclipse 文件名为 eclipse-java-luna-SR2-win32-x86_64.zip。

(1) 将 eclipse-java-luna-SR2-win32-x86_64.zip 文件解压后，双击 eclipse.exe 文件就可以启动 Eclipse。

(2) 每次启动 Eclipse 时，都需要设置工作空间，工作空间用来存储项目。可通过选择“Browse”按钮来选择一个存储项目的目录，如图 1-13 所示。通过选中“Use this as the default and do not ask again”选项屏蔽该对话框。

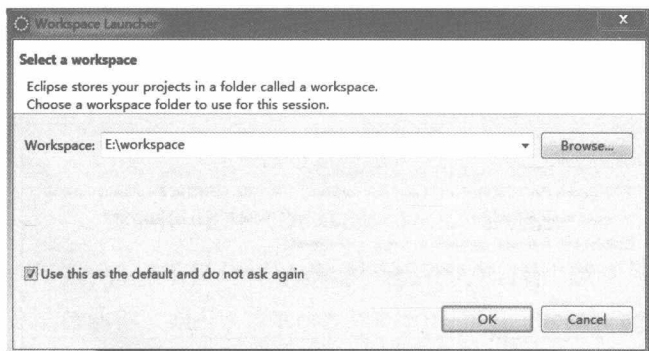


图 1-13 工作空间

(3) 单击“OK”按钮，若是初次使用，则出现 Eclipse 的欢迎界面，如图 1-14 所示。

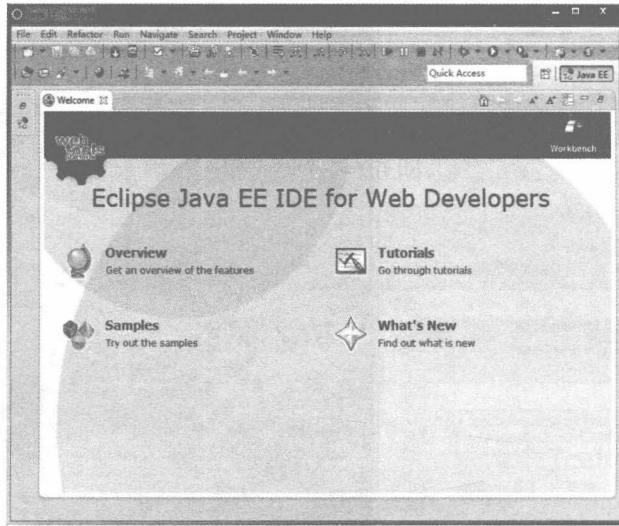


图 1-14 欢迎界面

1.4.4 安装 ADT 插件

Google 专门为 Eclipse 开发了一个插件(Android Development Tools),用它来辅助开发。安装 ADT 插件后,不仅可以联机调试,而且还能够模拟各种事件、分析程序性能等。Android 官网中提供了两种安装 ADT 插件的方法:一种是离线安装,另一种是在线安装。由于在线安装需要联网下载安装内容,所以这里只介绍离线安装方法。需要将其安装在 Eclipse 上,具体步骤如下。

(1) 启动 Eclipse 后,在主菜单上选择“Help”→“Install New Software”,此时弹出图 1-15 所示的“Install”对话框。

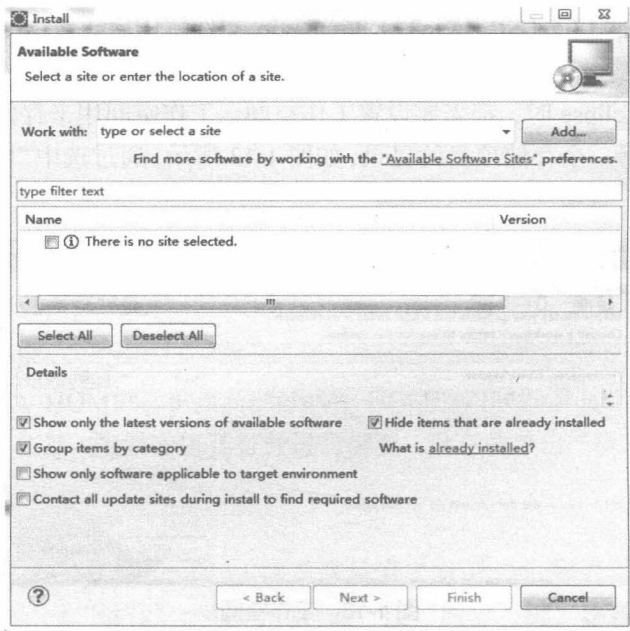


图 1-15 “Install”对话框

(2) 单击“Add”按钮，弹出图 1-16 所示的“Add Repository”对话框。

(3) 在“Name”文本框中输入 ADT，再单击“Archive”按钮，在打开的“Repository archive”对话框中，选择已经下载的 ADT 插件的离线安装包，单击“打开”按钮，返回到“Add Repository”对话框中，如图 1-17 所示。

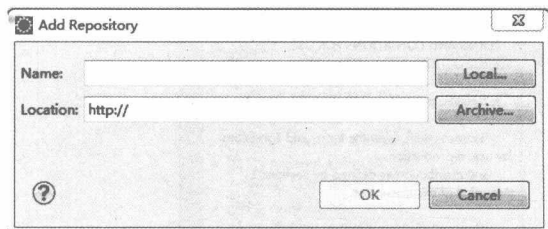


图 1-16 “Add Repository”对话框

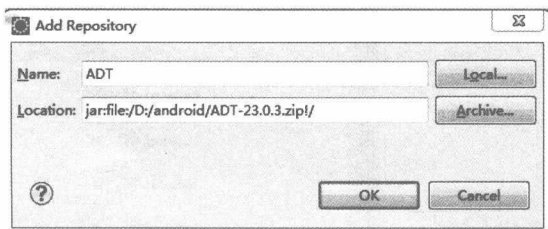


图 1-17 选择 ADT 插件的离线安装包的对话框

(4) 单击“OK”按钮，返回到“Install”对话框中，选中“Developer Tools”复选框，如图 1-18 所示。

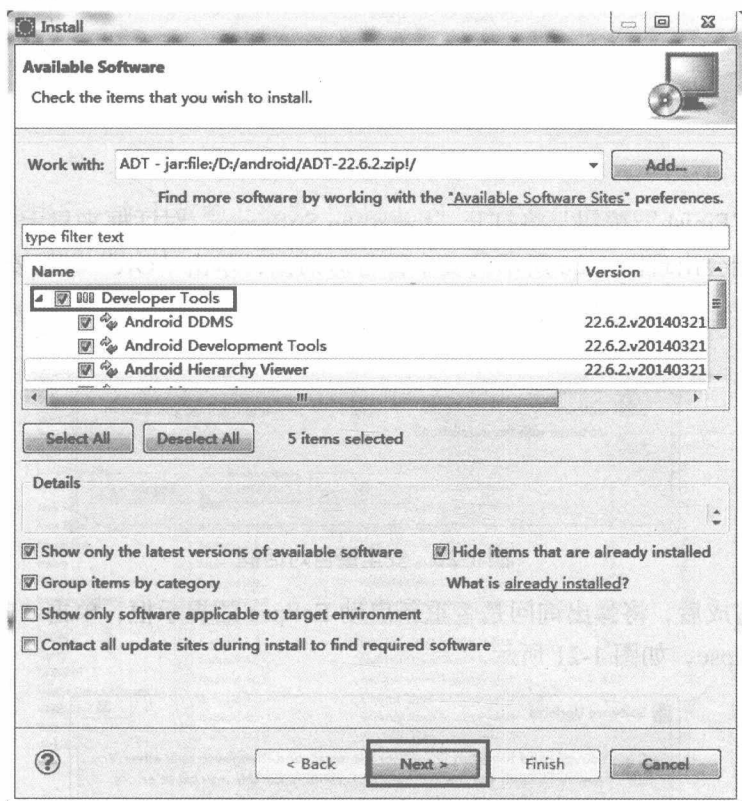


图 1-18 选中“Developer Tools”复选框

(5) 单击“Next”按钮，将显示插件详细信息，包括插件名称、版本号和 ID。单击“Next”按钮，将显示安装协议对话框。选中“I accept the terms of the license agreements”复选框，这时“Finish”按钮将变化为可用状态，如图 1-19 所示。