

21
世纪

高等学校计算机应用型本科规划教材精选



Android 移动应用 程序开发教程



朱凤山 编著
王慧芳 主审



清华大学出版社

21 世纪高等学校计算机应用型本科规划教材精选

Android 移动应用程序开发教程

朱凤山 编著

清华大学出版社
北京

内 容 简 介

本书主要介绍 Android 平台移动应用程序开发的知识,从基础知识开始讲解,由易入难,循序渐进,系统地介绍了 Android 应用程序开发中所用到的知识。注重引导学生掌握开发技巧,理解处理问题的思路,培养学生分析问题、解决问题的能力。本书可作为高等院校计算机、软件工程及相关专业的本、专科学生学习 Android 移动平台应用程序开发的教材,也可供该领域的教师、开发人员学习研究 Android 移动平台应用程序开发时参考。

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 移动应用程序开发教程/朱凤山编著. —北京:清华大学出版社,2014

21 世纪高等学校计算机应用型本科规划教材精选

ISBN 978-7-302-35978-4

I. ①A… II. ①朱… 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(2014)第 066053 号

责任编辑:索梅薛阳

封面设计:杨兮

责任校对:梁毅

责任印制:何芊

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 刷 者:三河市君旺印装厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:18.75

字 数:468 千字

版 次:2014 年 5 月第 1 版

印 次:2014 年 5 月第 1 次印刷

印 数:1~2000

定 价:34.50 元

产品编号:051835-01

前言

FOREWORD



Android 是一种基于 Linux 内核、开放源代码的操作系统,主要用于移动设备,如智能手机、平板电脑、电视等。来自互联网的统计数据显示,Android 已经成为目前使用最为广泛的移动操作系统,远超 Apple 公司的 iOS 和 Microsoft 公司的 Windows Phone。根据 Gartner 对智能手机操作系统占有市场份额的预期,到 2015 年 Android 操作系统的占有份额将达到 50% 左右,远高于其他操作系统。

对于学习 Java 编程语言的读者,Android 操作系统的出现,提供了新的学习方向。巨大的市场需求,提供了更多的机会,也急需更多的开发者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人群。如果你目前正处于这种状态下,本书比较适合你的选择。

本书在编写过程中,按照知识的逻辑关系分章,循序渐进、重点突出,对知识点的讲解与介绍做到尽量全面,并给出可以应用于何种场合的建议。对于重、难点知识,给出专门的演示项目,按步骤讲解实现方式。多数学习开发的读者在熟悉了语法知识之后,都想迫不及待地一展身手,编写一款自己的软件,虽然这是良好的学习习惯,也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,往往很难将功能实现,即使有参考代码和帮助文档,也会陷入代码海洋或文档风暴中,这样只会收到事倍功半的效果,而且,学习的积极性也会受到很大的打击。所以,对于初学者,建议选择功能单一、结构简单的项目。

全书所有章节讲解知识的方式统一,章节结构清晰,方便读者快速查询相关问题。每个章节开始都给出了该章的主要内容,列举出该章主要介绍的知识点。在介绍内容时,根据不同知识点的具体情况,介绍知识的分类、周边信息并总结功能实现的步骤。书中经常涉及以下符号,读者应了解其含义:

- 表明知识点涉及内容的划分或实现步骤的说明。如对某个类中常用方法的说明,实现某个功能可以采用的方式有哪些。该符号所对应的知识点都需要掌握、熟记,能够达到应用的水平。
- 关键功能代码。该符号对应一段能够实现某个功能的关键代码,考虑到篇幅问题,多数情况下,它们并不是一段完整的代码,如果需要阅读项目的所有源代码,可查阅本书配套电子资料。
- 对某些知识在运用到项目中的建议,或者在使用某些知识时应该注意的事项。

本书配套电子资料中含有本书中所列全部项目(Workspace 是 BookDemo),读者可以将整个工作空间都引入 Eclipse 中。不同章节项目的命名以 part××区分,同一个章节中不同项目的命名以 part××_×区分。读者应特别留意命名为 part××_×_×的项目,这些项目是某些 Android 项目所依赖的 Web 端程序,主要是第 10 章的项目。配套资料中的



II Android 移动应用程序开发教程

Android 项目在开发时采用 Eclipse ADT4.2、SDK4.2,运行的目的平台最低是 Android 2.3.3。配套资料中的 Web 项目在开发时采用 MyEclipse 9.1,这些项目需要部署在服务器中(如 Tomcat 6.0)才能运行。

全书共分为 10 章。第 1 章介绍 Java 语言的基础知识;第 2 章介绍 Android 开发环境,Android 项目的结构,Android 项目的调试与发布,并详细介绍了 Activity 的使用;第 3 章介绍 Android 平台 android.widget 包中基本控件的使用,包括 Form Widgets 控件、TextFields 控件、布局管理器、Image 和 Media 控件、Time 和 Date 控件等;第 4 章介绍 Android 平台高级控件的使用,包括 ListView 与适配器的使用、ExpandableListView、GridView、ScrollView 和 HorizontalScrollView、SlidingDrawer、TabHost 和 TabSpec、Gallery 和 ImageSwitcher;第 5 章介绍 Android 平台系统组件的使用,包括 Menu、自定义 Dialog、Notification、ActionBar 的功能解析和布局新方式 Fragment;第 6 章介绍四大组件 Activity、Service、BroadcastReceiver、ContentProvider,以及 Intent 与 IntentFilter;第 7 章介绍 2D 游戏开发的基础知识,包括 View 与 SurfaceView 的使用、Canvas 和 Paint 介绍、如何绘制游戏元素、屏幕坐标与屏幕事件等;第 8 章介绍多媒体操作的内容,包括 MediaPlayer、SoundPool、VideoView 的使用;第 9 章介绍 Android 平台数据存储的三种方式,包括 SharedPreferences 与 Editor 的使用,I/O 流与读写 SD 卡,嵌入式数据库 SQLite 的使用;第 10 章是 Android 平台网络编程的内容,包括 TCP 通信与 Socket 应用、URL 访问网络资源、HttpClient 的应用与 Webservice 的应用、XML 解析与 JSON 解析。

作为 developer.android、CSDN、51CTO、eoeandroid、机锋开发者等技术论坛和社区的忠实用户和学习者,在本书的编写过程中,作者从中受益匪浅,也建议读者在遇到学习问题时,可以向专业技术论坛或社区求助。在本书完成之际,特别要感谢王慧芳教授、王志军教授给予的指导和帮助,感谢桑婧、王慧、徐峰、王新峰和新锐 IT 工作室的成员给予的启发和帮助,也要感谢张新芳、朱思齐的大力支持。

由于作者学术与经验的欠缺,在本书的结构、知识点与难点的选择和解析过程中,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 tj_zhufengshan@163.com,只要有时间,作者当尽量给每个人回信。

朱凤山

2014 年 1 月

目 录

CONTENTS



第 1 章 开发语言与开发环境	1
1.1 Java 背景知识	1
1.2 开发环境配置	2
1.2.1 Java SDK 的安装、配置与测试	3
1.2.2 Eclipse 的安装与测试	4
1.2.3 搭建 Android 开发环境	6
1.3 Java 数据类型与运算符	10
1.3.1 基本数据类型	10
1.3.2 运算符	11
1.3.3 不同数据类型间的转换	12
1.3.4 引用数据类型	12
1.4 Java 基本流程控制语句	12
1.4.1 分支控制语句	13
1.4.2 循环控制语句	14
1.4.3 辅助语句	14
1.4.4 其他控制语句	15
1.5 Java 语言的特点	15
1.5.1 强类型	15
1.5.2 完全面向对象	15
1.5.3 多线程	16
1.5.4 可移植性	16
1.5.5 其他特点	17
习题	17
第 2 章 初识 Android 平台	18
2.1 Android 平台介绍	18
2.1.1 Android 平台的由来	18
2.1.2 Android 历届版本	19
2.1.3 Android 平台的特征	20
2.2 Android 应用程序的测试设备	23
2.2.1 虚拟机的创建	24



2.2.2	虚拟机的使用	26
2.2.3	真机测试	28
2.3	解析 Android 项目结构	29
2.3.1	创建 Android 项目	29
2.3.2	Android 项目结构介绍	31
2.3.3	运行结果分析	36
2.4	应用程序与 Activity	38
2.4.1	Activity 介绍	39
2.4.2	Activity 的生命周期	39
2.4.3	Intent 与 Intent Filter	45
2.5	程序调试与应用发布	45
2.5.1	Console 与 LogCat	45
2.5.2	断点调试	47
2.5.3	打包发布与签名	47
	习题	49
第 3 章	基本控件与布局管理器	50
3.1	widget 包与控件	50
3.1.1	控件的分类	51
3.1.2	UI 的编辑方式	52
3.1.3	控件的属性	53
3.2	Form Widgets	56
3.2.1	TextView	56
3.2.2	Button	56
3.2.3	ToggleButton	58
3.2.4	RadioButton 与 RadioGroup	59
3.2.5	CheckBox	60
3.2.6	CheckedTextView	60
3.2.7	ProgressBar	60
3.2.8	SeekBar	62
3.2.9	Spinner	63
3.2.10	QuickContactBadge	64
3.2.11	RatingBar	65
3.3	TextFields	66
3.4	布局管理器	67
3.4.1	LinearLayout	67
3.4.2	RelativeLayout	69
3.4.3	TableLayout	71
3.4.4	FrameLayout	73

3.4.5	AbsoluteLayout	74
3.5	Image 和 Media	74
3.5.1	ImageView 与 BitmapFactory	74
3.5.2	ImageButton	77
3.6	Time 和 Date	78
3.6.1	TimePicker 和 DatePicker	78
3.6.2	Chronometer	80
3.6.3	AnalogClock 与 DigitalClock	81
习题	82
第 4 章	高级控件与数据适配器	83
4.1	ListView 与适配器	83
4.1.1	ArrayAdapter 适配器	84
4.1.2	SimpleAdapter 适配器	85
4.1.3	带有事件监听的 ListView	87
4.1.4	自定义适配器	90
4.2	ExpandableListView	93
4.3	GridView	95
4.4	ScrollView 和 HorizontalScrollView	97
4.5	SlidingDrawer	97
4.6	TabHost 和 TabSpec	100
4.7	Galley 和 ImageSwitcher	102
4.7.1	简单 Gallery	102
4.7.2	图片切换	104
习题	106
第 5 章	使用系统组件	108
5.1	Menu	108
5.1.1	OptionsMenu	108
5.1.2	SubMenu	111
5.1.3	ContextMenu	112
5.2	Dialog	114
5.2.1	AlertDialog 与 Builder	114
5.2.2	ProgressDialog	121
5.2.3	DatePickerDialog 和 TimePickerDialog	122
5.2.4	自定义布局对话框	124
5.2.5	Dialog 样式的 Menu	126
5.3	通知	129
5.3.1	Toast	129



5.3.2	Notification	130
5.3.3	定制 Notification	134
5.4	ActionBar	136
5.4.1	启用 ActionBar	136
5.4.2	处理 Action 菜单	137
5.4.3	启用应用程序图标	139
5.4.4	添加可交互视图	140
5.4.5	标签导航	141
5.4.6	下拉导航	143
5.5	Fragment	144
5.5.1	创建并使用 Fragment	144
5.5.2	Fragment 生命周期	149
5.5.3	管理 Fragment	150
	习题	150
第 6 章	Android 四大组件	151
6.1	Activity	151
6.2	Service	152
6.2.1	新建 Service	152
6.2.2	Service 的生命周期	155
6.2.3	Local Service 和 Remote Service	156
6.3	BroadcastReceiver	164
6.3.1	广播接收器的注册	164
6.3.2	广播的分类	168
6.3.3	权限与系统广播	172
6.4	ContentProvider	175
6.4.1	使用 ContentProvider	175
6.4.2	Uri	177
6.4.3	ContentProvider 基本操作	178
6.5	Intent 与 IntentFilter	181
6.5.1	Component、Action 与 Category	181
6.5.2	Data 与 Type 属性	185
6.5.3	Extra 与 Flag 属性	186
	习题	188
第 7 章	2D 游戏开发	189
7.1	游戏开发基础	190
7.1.1	开发前的思考	190
7.1.2	关于刷屏	190

7.1.3	屏幕坐标系	190
7.1.4	横屏和竖屏	191
7.1.5	全屏操作	192
7.2	绘制游戏元素	193
7.2.1	View 视图	193
7.2.2	Canvas 画布	194
7.2.3	Paint 画笔	200
7.2.4	SurfaceView 视图	201
7.3	游戏元素的控制	203
7.3.1	按键监听	203
7.3.2	触屏监听	205
7.3.3	线程	206
7.4	位图的使用	208
7.4.1	创建位图	208
7.4.2	位图的操作	209
7.4.3	9patch 编辑器	209
7.5	动画	212
7.5.1	tweened animation	212
7.5.2	frame-by-frame animation	217
7.5.3	自定义动画	218
7.5.4	剪切区动画	219
7.6	游戏元素的碰撞	221
7.6.1	矩形碰撞	221
7.6.2	圆形碰撞	222
7.6.3	Region	222
	习题	223
第 8 章	音频与视频的使用	224
8.1	MediaPlayer	224
8.1.1	创建 MediaPlayer	224
8.1.2	设置播放文件	225
8.1.3	播放器的控制	226
8.1.4	播放器的监听器	227
8.2	SoundPool	227
8.3	VideoView	229
8.4	MediaRecorder	231
8.4.1	录制声音	232
8.4.2	录制视频	233
	习题	236

第 9 章 数据的存储	237
9.1 SharedPreferences 读写 XML 文件	237
9.1.1 SharedPreferences 基本操作	237
9.1.2 Editor 写入数据	238
9.2 使用 I/O 读写文件	240
9.2.1 读写应用程序中的文件	240
9.2.2 读写 SD 卡中的文件	242
9.3 SQLite 数据库	245
9.3.1 SQLiteDatabase	245
9.3.2 数据库的基本操作	247
9.3.3 SQLite 管理工具	251
9.3.4 SQLiteOpenHelper	252
习题	253
第 10 章 网络编程	254
10.1 基于 TCP 的通信	254
10.1.1 TCP 与 Socket 编程	254
10.1.2 ServerSocket 与 Socket	255
10.2 URL 获取网络资源	258
10.2.1 URL 介绍	258
10.2.2 URLConnection 与 HttpURLConnection	262
10.2.3 Get 请求与 Post 请求	264
10.2.4 HttpClient	268
10.3 使用 Web Service	273
10.3.1 调用 Web Service	273
10.3.2 解析 XML	277
10.3.3 航班信息查询	278
10.3.4 解析 JSON	283
习题	287
参考文献	288



开发语言与开发环境

本章主要内容:

- Java 背景知识;
- 开发环境配置;
- Java 基本数据类型;
- Java 基本控制语句;
- Java 语言的特点。

1.1 Java 背景知识

Java 起源于 20 世纪 90 年代初 Sun 公司(于 2009 年被 Oracle 公司收购)的一个叫 Green 的项目,该项目的目标是开发嵌入家用电器的分布式软件系统,使电器更加智能和通用(避免针对不同芯片重复编码,即让代码在不同芯片上都可以正常运行)。

1998 年 12 月 4 日,JDK 1.2 隆重发布,标志着 Java 2 平台的诞生。Sun 公司在 Java 1.2 版以后将 JDK 1.2 改名为 J2SDK,将 Java 改名为 Java 2。在 1999 年 Sun 公司还将 Java 2 平台分为三大块:J2EE、J2SE、J2ME,具体说明见表 1.1。

表 1.1 Java 2 的三大平台

名称	说 明
J2EE	Java 2 Enterprise Edition(企业版),适用于服务器,目前已成为企业运算、电子商务等领域的热门技术
J2SE	Java 2 Standard Edition(标准版),适用于一般的计算机,开发 PC 上的应用软件
J2ME	Java 2 Micro Edition(微型版),适用于手持设备,进行应用开发,如手机游戏、名片管理等

在今后的讲解和介绍中,将以 J2SE 6.0 为平台,讨论 Java 语言程序基础知识。J2SDK 1.6 的标准版又名 Java SE 6。读者无须再纠结于 Java 的哪个开发版本,尤其是刚刚接触的读者,无须关注各版本间的差别,建议统一选择 J2SE 6.0 作为自己的学习和开发平台,对于目前最新的 J2SE 7.0 感兴趣的读者可以自己尝试。

不仅平台发生变化,随着供应商的不同,Java 的 API(Application Programming Interface,应用程序编程接口,供程序员调用)被分为三大类,Google 公司对 Android 平台提供了第 4 类 API,这些 API 在后续的学习中都会有所涉及。

- Java Core API: 由 Sun 公司制定的基本的 API,包名是 Java. ,所有的 Java 平台都应该提供。
- Java Optional API: 由 Sun 公司制定的扩充 API,包名是 Javax. ,Java 平台可以有选择地提供。
- 特殊 API: 由特殊厂商或者组织提供的 API,包名是 org. 。
- Android 平台所采用的由 Google 公司提供的 API,包名是 android. 。

时至今日,J2SE 已经发展为一个覆盖面广、效率高、易用性强的技术平台,其体系结构如图 1.1 所示。截至目前,读者不要企图搞明白图中的所有技术,在以后的讲解中,也不可能介绍全部的知识,读者要明白一点,J2SE 6.0 已经非常成熟和完善了,提供了强大而丰富的 API,它足以让我们很轻松地完成基于该平台上的所有软件开发工作。

Java Language		Java Language								
Tools & Tool APIs	java	javac	javadoc	apt	jar	javap	JPDA	jconsole		
	Security	Int'l	RMI	IDL	Deploy	Monitoring	Troubleshoot	Scripting	JVM TI	
Deployment Technologies	Deployment			Java Web Start				Java Plug-in		
	AWT			Swing			Java 2D			
User Interface Toolkits	Accessibility		Drag n Drop		Input Methods		Image I/O	Print Service	Sound	
Integration Libraries	IDL	JDBC~		JNDI~		RMI	RMI-IIOP		Scripting	
Other Base Libraries	Beans		Intl Support		I/O	JMX	JNI		Math	
	Networking		Override Mechanism		Security	Serialization	Extension Mechanism		XML JAXP	
lang and util Base Libraries	lang and util	Collections	Concurrency Utilities			JAR		Logging	Management	
	Preferences API	Ref Objects	Reflection			Regular Expressions		Versioning	Zip	Instrument
Java Virtual Machine	Java Hotspot~ Client VM					Java Hotspot~ Server VM				
Platforms	Solaris~			Linux			Windows		Other	

图 1.1 Java 6.0 API 结构

通过上面的介绍读者应该清楚一点,Java 语言比较适合开发互联网应用程序、移动应用程序,如果目标是这个方向的,那么选择 Java 语言是比较合适的。下面给出选择编程语言的几条建议。

- 编程语言目前或将来是否占有主导地位,这样可以保证所开发的产品具有广泛的市场。
- 编程语言是否支持重用,以便于提高软件的开发效率。
- 类库和开发环境是否比较成熟。

1.2 开发环境配置

确定要学习 Java 语言,首先要准备开发环境,就好比练习刀枪需要先准备场地一样。下面分别介绍两种开发环境,一种是 Java SDK 自带的,效率比较低,另一种是集成的开发环境(Integrated Development Environment, IDE),可以很快地组织、编写和测试代码。

1.2.1 Java SDK 的安装、配置与测试

目前使用较多的 SDK 版本是 6.0,在后续编程过程中,建议读者使用 SDK 6.0,便于知识点同步。下面分步介绍如何安装和配置该环境。

(1) 获取安装包。SDK 安装包可以去 Oracle 的官方网站下载,也可以去百度或 Google 网站搜索。需要注意的是,获取的安装文件是否是 Windows 平台兼容的。比如 jdk-6u1-windows-i586-p.exe,就是可以安装在 Windows 平台上的。

(2) 安装包里除了包含 Java 语言编译器、调试器以及演示程序以外,一般还会包含 Java 程序的运行环境(Java Runtime Environment,JRE)。JRE 是某一平台运行 Java 程序的软件环境,包括虚拟机 JVM(Java 程序可以跨平台运行的基础)和核心类库等。安装过程中,会询问是否安装 JRE,选择安装就可以了,建议默认安装到 C 盘(文件大约 70MB)。

(3) 配置 JDK。安装完成后,需要对其进行配置。右击“我的电脑”,选择“属性”命令,在“系统属性”窗口中打开“高级”选项卡,单击最下面的“环境变量”按钮,弹出“环境变量”编辑窗口。在“系统变量”面板中单击“新建”按钮,创建如表 1.2 所示的三个环境变量(JDK 默认安装情况下)。

表 1.2 JDK 环境变量

名称	值	说明
JAVA_HOME	C:\Program Files\Java\jdk1.6.0_13	JDK 根目录
Classpath	.;%JAVA_HOME%\lib;	lib 类库路径,注意有个“.”,多个值用“;”隔开
Path	%JAVA_HOME%\bin;path	bin 目录路径

(4) 测试安装。配置好环境变量后,写一个简单的程序以作测试。建议在 D 盘或 E 盘中建立一个文件夹,命名为 BookDemo,再建立 test01 子文件夹。在路径 E:\BookDemo\Part01 中新建一个文本文件,并复制以下代码。

代码 01-1

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

将该文本文件改名为 FirstDemo.java(如没有后缀名,需要在文件夹选项中开启显示常见文件后缀名),与代码中 class 后面的类名一致。

在“运行”界面中输入“cmd”,进入命令行界面,输入“E:”,即可进入 E 盘,输入“cd BookDemo\Part01”进入代码所在文件夹。输入命令“javac FirstDemo.java”,进行编译,如无错误,该文件夹下会生成一个名为 FirstDemo.class 的文件,如图 1.2 所示。其中.java 文件是源代码文件,刚刚生成的.class 文件是编译后的文件,可以直接运行。



图 1.2 编译后的文件

在命令行界面输入命令“java FirstDemo”，显示“Hehe”，表明代码运行正常，环境变量配置正确，全部过程如图 1.3 所示。

```
E:\BookDemo\part01>javac FirstDemo.java
E:\BookDemo\part01>java FirstDemo
Hehe
E:\BookDemo\part01>
```

图 1.3 代码 01-1 的编译和运行

如果在测试过程中，没有得到上面的结果，请按以下几条原因排查。

- 代码并非复制，而是自己动手输入的（如果真是这样做的，恭喜你已获取学习程序语言的秘籍了），请对照代码 01-1，检查是否有拼写错误，Java 语言对大小写是敏感的，务必注意这一点。
- 环境变量配置有误，请对照前面的介绍，仔细检查环境变量的配置。
- 文件命名是否正确，要求文件名称与代码中 class 后面的类名一致，且后缀为 .java。

1.2.2 Eclipse 的安装与测试

借助 JDK 自带的 javac 和 java 命令，虽然可以编写程序，但是效率比较低。下面介绍如何借助集成化的开发工具编写代码。支持 Java 开发的集成开发工具有很多，如 JBuilder、Netbeans、WebLogic、Eclipse。在以后的开发过程中，主要使用 Eclipse 开发环境，这也是目前比较流行且成熟的开发工具。

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。最初是 IBM 公司的产品，后又贡献给开源组织 Eclipse.org。2003 年 3 月，Eclipse 2.1 一经发布，以其友好的界面和强大的功能，吸引了众多大公司加入到 Eclipse 平台的发展上来，目前许多实际开发中应用的开源框架如 Spring、Struts、Hibernate 等都会附带提供 Eclipse 的插件支持。

安装、使用 Eclipse 比较简单，可参看以下步骤进行。

(1) 确保 JRE 已正确安装。

Eclipse 需要 Java 运行环境的支持，必须先安装 JRE，如果已经完全安装过 JDK，此步骤可以省略。

(2) 获取 Eclipse 安装包。

安装包可以在 <http://www.eclipse.org/downloads/> 网站下载，由于版本众多，读者要仔细阅读说明，而且要注意 32 位操作系统和 64 位操作系统是不同的。后面讲述使用的是 Eclipse Java EE IDE for Web Developers，版本为 Helios Release。

下载与操作系统匹配的安装包后，直接解压到相应目录即可，Eclipse 是完全绿色的，无须安装。进入解压后的路径，将 eclipse.exe 发送到桌面快捷方式，方便以后打开。

(3) 测试 Eclipse。

第一次打开该软件会选择工作文件夹(便于所有创建的项目和代码文件的存放),直接选择前面创建的 workspace 文件夹即可,Eclipse 打开后会显示欢迎界面,直接关闭也可,现在不对该软件一一介绍,随着学习的深入,会慢慢了解到这款软件的强大之处。软件的布局和常用的办公软件差不多,上面是菜单栏和快捷工具栏,中间区域是主要部分,左边是 Package Explorer,以后新建的项目都会罗列在此,中间是主要编辑区域,用于完成代码的编辑,右边是 Outline,显示所编辑区域的类信息。下面还有一些其他视图,这些窗口视图可以在菜单中选择 Window→Show View 命令。目前会用到的是 Console(控制台)视图,用以显示输出信息和输入数据。

选择 File→New→Java Project 命令,输入项目名称“BookDemo”,单击 Finish 按钮,完成创建新项目。展开此项目会有两部分,上面是 Src 文件夹,下面是 JRE。程序代码当然要写在 Src 文件夹中,但是,一般不会直接在 Src 下面建立代码文件,而是先创建一个 package(包),然后将代码文件创建在包中,当然,包下面可以继续创建包,这样的好处是用包来管理代码文件,比较方便,功能类似的代码放在同一个包中,也便于查找。

选择 Src 文件夹,右击,选择 New→Package 命令,新建一个包,输入包名(建议小写),再选择此包,选择 New→class 命令,新建一个类,输入类名。全部完成后,Package Explorer 视图中应该如图 1.4 所示。

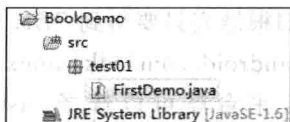


图 1.4 新建 FirstDemo 视图

代码 01-2

```
package test01; //包名
public class FirstDemo { //自动生成的类,class 关键字用于定义类,class 前是修饰词
    //main 方法是代码的主方法,如果代码要运行,需要 main 方法
    public static void main(String args[]){
        System.out.println("Hehe"); //实现向 Console 输出字符串 Hehe
    }
}
```

输入代码 01-2,基本和代码 01-1 一致,只是增加了包名的声明(Eclipse 自动添加),前面已经介绍了代码是用包的方式来管理的,希望读者在编写代码时都放在不同的包中。代码编写完成后,在代码编写区域单击右键,选择 Run as→Java Application 命令,如在 Console 视图输出字符串“Hehe”,表明 Eclipse 可以正常工作。关于 Eclipse 的使用和 Java 代码的编写,应遵循以下几条建议。

- 项目的命名、包的命名、类的命名尽量使用字母和数字的组合,且包名一般都小写,类名每个单词的首字母大写(驼峰表示)。
- 书写代码时注意缩进,错落有致,请读者品味以上两段代码的书写风格。
- 代码应该有注释语句,Java 中常用的注释方式有三种,分别是 `//`、`/* */`、`/** */`

- * /。后两种可以用于多行注释,后面的代码中会用到。
- 在编写代码的同时,Eclipse 已经编译好代码了(编译好的代码放在 bin 目录中,bin 目录是与 Src 对应的一个目录),所以可以直接运行,不需要向前面介绍的那样先 javac,然后 java,直接运行即可。
- 借助 Eclipse 编写代码,需要先建立一个 class,增加一个 main 方法,然后将要运行的代码写在 main 方法中(main 方法由 JVM 调用)。

1.2.3 搭建 Android 开发环境

到此为止,开发 Java 应用程序的环境已经搭建完毕,在此基础上下面介绍如何搭建开发 Android 应用程序的环境。整个过程稍显烦琐,需要先下载 Android 的 SDK(Software Development Kit,软件开发工具包),配置环境变量,然后给 Eclipse 安装(“升级”更准确)插件 ADT(Android Development Tools),并指定 SDK 的位置,最后创建 Android 虚拟设备(Android Virtual Device,AVD),即虚拟 Android 系统手机。下面给出详细的安装与配置步骤。

1. 下载 SDK

SDK 的版本更新得比较快,获取的方式也在改变,以下描述的获取 SDK 的过程是当时最新的,但无须按部就班地进行,归根结底只要得到 SDK 就可以。Android 开发社区,下载 SDK 的网址是 <http://developer.android.com/sdk/index.html>,页面如图 1.5 所示,选择合适的开发平台,如果是 Windows 平台就可以选择 installer_r12-windows.exe,这也是 Android 推荐项。具体安装事宜可以阅读该网站的 Installing the SDK,比如系统的需求、软件的需求等。如果采用本书的开发环境,则无须阅读,可以直接向下进行。

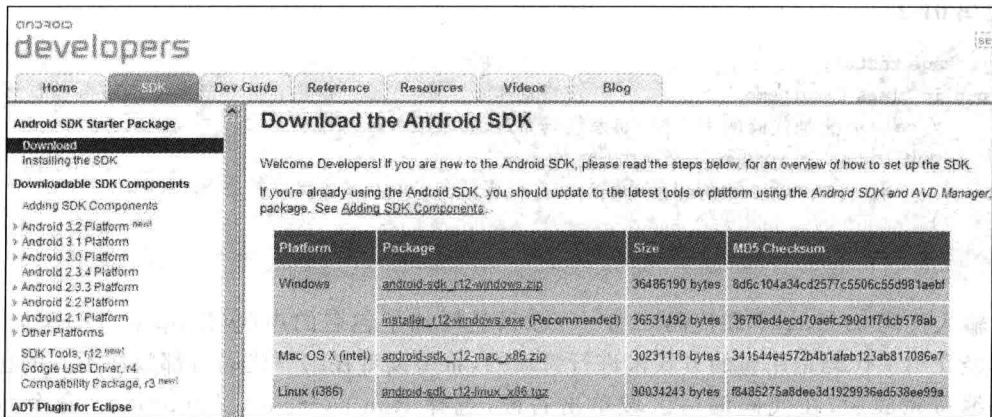


图 1.5 SDK 下载页面

安装结束后,会打开 Android SDK and AVD Manager 窗口,开始更新,选择全部更新即可,更新有时会比较慢,取决于网速,更新界面如图 1.6 所示。更新完成后,将安装路径中的 tools 文件夹增加到环境变量 path 中,在本书中是 E:\Android\android-sdk\tools。检验 SDK 是否安装成功可以在“运行”界面中输入“cmd”,打开命令行窗口,输入“android -h”,如