

校企(行业)合作
系列教材

Java CHENGXU
SHEJI JIAOCHENG

Java程序 设计教程

主 编：黄朝辉 副主编：陈建辉



厦门大学出版社 国家一级出版社
XIAMEN UNIVERSITY PRESS 全国百佳图书出版单位

校企(行业)合作
系列教材

Java程序设计教程

主 编：黄朝辉

副主编：陈建辉

参 编：林建兵 张勇敢 杨宗强



厦门大学出版社 国家一级出版社
XIAMEN UNIVERSITY PRESS 全国百佳图书出版单位

图书在版编目(CIP)数据

Java 程序设计教程/黄朝辉主编. —厦门:厦门大学出版社,2018.1
ISBN 978-7-5615-6817-0

I. ①J… II. ①黄… III. ①JAVA 语言-程序设计-高等学校-教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 318000 号

出版人 郑文礼
责任编辑 陈进才
封面设计 蒋卓群
技术编辑 许克华

出版发行 厦门大学出版社
社 址 厦门市软件园二期望海路 39 号
邮政编码 361008
总 编 办 0592-2182177 0592-2181406(传真)
营销中心 0592-2184458 0592-2181365
网 址 <http://www.xmupress.com>
邮 箱 xmupress@126.com
印 刷 厦门市金凯龙印刷有限公司

开本 787mm×1092mm 1/16
印张 14.75
插页 1
字数 358 千字
印数 1~2 000 册
版次 2018 年 1 月第 1 版
印次 2018 年 1 月第 1 次印刷
定价 39.00 元

本书如有印装质量问题请直接寄承印厂调换



厦门大学出版社
微信二维码



厦门大学出版社
微博二维码

前 言

自 Sun 公司 1995 年正式推出 Java 语言以来,Java 语言以其独到的特性——简单易学、面向对象、跨平台、安全、稳健、支持分布、多线程和无线应用等,历经 20 多年逐步发展成为当今流行的一种纯面向对象、网络编程首选的程序设计语言。Java 不仅是一门受欢迎的程序设计语言,也已形成一门专门的技术,从智能卡应用、手持式电子消费类产品应用、桌面应用到大型复杂的企业级应用,都有 Java 的活跃身影。Java 作为软件开发的一种革命性技术,已成为网络时代世界信息技术的主流之一。

随着社会对 Java 人才需求的增加,国内各高校计算机、电子信息 and 通信等理工科专业都先后开设了 Java 类课程,Java 语言课程日趋普及。本教材旨在满足高等院校 Java 类课程的教学与实践需要,以 Java 编程者的入门和提高编程能力为着手点,从 Java 语言基础、面向对象方法、网络环境下的应用与开发、高级应用 4 个层面,全面介绍了 Java 语言规范、Java 面向对象的编程机制、Java 基本类库、图形用户界面设计、Java 多线程机制、Java 网络编程技术、Java 与数据库连接等。

本书在内容安排上力求通俗易懂、由易到难、循序渐进、图文并茂,并配合大量实用、经典的编程实例,以及每章后的思考题和上机练习题。即使是没有编程经验的新手,通过本书的阅读和学习也可以较快地掌握 Java 编程技术,并将本书介绍的大量应用实例运用到实际开发中去。本书可作为高等院校 Java 语言课程的教材,也可供从事网络技术、软件开发的专业人员参考,或适合于编程开发人员培训、广大 Java 爱好者参考与自学使用。

本书编著者队伍由大学的计算机教师和计算机公司的程序设计员组成,他们有丰富的教学 and 实践经验。本书由黄朝辉(莆田学院)任主编,陈建辉(莆田学院)任副主编,林建兵(莆田学院)、张勇敢(莆田学院) and 杨宗强[中软国际(厦门)公司]参与部分章节的编写工作,全书由黄朝辉统稿。

本书在编写过程中,得到了福州大学博士生导师叶东毅教授、厦门大学嘉庚学院张思民教授的精心指导,他们提出的宝贵意见使本书更趋完善。此外,本书的编写与出版也得到了莆田学院、中软国际(厦门)公司和厦门大学出版社的大力支持和帮助,在本书即将出版之际一并表示衷心的感谢。

本书的编写主要依据 Java SE 标准,融入了编著者们多年程序设计的教学经

验和使用 Java 开发应用的经验。写作过程中所参考的书籍资料,书中恕不一一注明出处,这些资料源自众多的大学、研究机构、商业团体以及一些研究 Java 编程的个人,对他们推动 Java 编程的应用和发展谨此致谢,其原文版权属于原作者,特此声明。

由于本书编写时间和编者自身水平有限,书中难免存在不妥之处,诚恳希望广大读者提出宝贵的意见和建议,以使本书的质量得到进一步的提高。

编著者

2017 年 10 月

目 录

第 1 章 Java 概述	1
1.1 Java 发展简史与特点	1
1.1.1 Java 发展简史	1
1.1.2 Java 特点	2
1.2 Java 开发与运行环境	3
1.2.1 Java 平台	3
1.2.2 JDK 下载与安装	4
1.2.3 Java 工具集	11
1.2.4 Eclipse 下载与安装	11
1.3 Java 程序简介	13
1.3.1 Java 应用程序	13
1.3.2 Java 小应用程序	16
本章小结	18
习题	19
第 2 章 Java 编程基础	20
2.1 Java 标识符、关键字与注释符	20
2.1.1 Java 标识符	20
2.1.2 Java 关键字	21
2.1.3 Java 注释符	21
2.2 Java 数据类型、常量与变量	21
2.2.1 Java 数据类型	21
2.2.2 常量与变量	23
2.3 运算符与表达式	28
2.3.1 算术运算符与算术表达式	28
2.3.2 关系运算符与关系表达式	29
2.3.3 逻辑运算符与逻辑表达式	29
2.3.4 赋值运算符与赋值表达式	30
2.3.5 复合赋值运算符与复合赋值表达式	31
2.3.6 条件运算符与条件表达式	31
2.3.7 位运算符与位运算表达式	31
2.3.8 其他运算符	33

2.3.9 运算符的优先级与结合性	34
2.3.10 数据类型的转换	35
2.4 程序控制语句	38
2.4.1 选择结构程序控制语句	38
2.4.2 循环结构程序控制语句	47
2.4.3 跳转控制语句	52
2.5 数组与字符串	54
2.5.1 一维数组	54
2.5.2 二维数组	56
2.5.3 数组应用举例	58
2.5.4 字符串	63
本章小结	68
习题	68
第3章 Java 面向对象程序设计	69
3.1 面向对象程序设计基础	69
3.1.1 对象与类的基本概念	69
3.1.2 面向对象程序设计的特性	71
3.1.3 面向对象程序设计的优势	71
3.2 类	71
3.2.1 类的定义	72
3.2.2 成员变量	73
3.2.3 成员方法	74
3.3 对象	75
3.3.1 创建对象	76
3.3.2 使用对象	76
3.3.3 释放对象	78
3.3.4 构造方法初始化对象	79
3.4 类的封装	83
3.4.1 访问控制修饰符	83
3.4.2 静态修饰符 static	86
3.5 继承与多态	87
3.5.1 子类的定义	88
3.5.2 成员变量的隐藏	89
3.5.3 成员方法的重写与重载	90
3.5.4 子类的构造方法	92
3.5.5 最终类与抽象类	94
3.5.6 多态性	98

3.6 接口与包	98
3.6.1 接口的定义	98
3.6.2 接口的实现	99
3.6.3 包的创建与应用	101
3.7 常用类	103
3.7.1 Math 类	103
3.7.2 Random 类	104
3.7.3 Arrays 类	106
3.7.4 Date 类、Calendar 类与 SimpleDateFormat 类	109
3.8 异常处理	113
3.8.1 什么是异常	113
3.8.2 Java 异常处理机制	115
3.8.3 自定义异常	119
本章小结	121
习题	121
第4章 图形用户界面设计	122
4.1 AWT 和 Swing	122
4.1.1 AWT	122
4.1.2 Swing	122
4.2 容器与组件	124
4.2.1 容器	124
4.2.2 组件	125
4.2.3 内容窗格	125
4.3 布局管理器	127
4.3.1 BorderLayout 布局管理器	127
4.3.2 FlowLayout 布局管理器	129
4.3.3 GridLayout 布局管理器	130
4.3.4 CardLayout 布局管理器	132
4.3.5 不使用布局管理器	135
4.4 事件机制	137
4.4.1 事件处理三要素	137
4.4.2 监听器	138
4.4.3 适配器	141
4.5 常用组件	143
4.5.1 按钮	143
4.5.2 文本框	146
4.5.3 菜单	148

4.5.4 Java 基本绘图	152
本章小结	155
习题	155
第 5 章 输入/输出流	156
5.1 输入输出基本概念	156
5.1.1 流的概念	156
5.1.2 输入输出流类概述	157
5.2 字节流	158
5.2.1 InputStream	158
5.2.2 OutputStream	159
5.2.3 字节流读写文件	160
5.3 字符流	161
5.3.1 Reader	161
5.3.2 Writer	162
5.3.3 字符流读写文件	162
本章小结	163
习题	164
第 6 章 多线程	165
6.1 线程概述	165
6.2 线程的创建	166
6.2.1 Thread 类与 Runnable 接口	166
6.2.2 继承 Thread 类创建线程	167
6.2.3 实现 Runnable 接口创建线程	169
6.2.4 两种实现多线程方式的对比分析	170
6.3 线程的生命周期及状态转换	173
6.3.1 线程的生命周期	173
6.3.2 线程的优先级与调度	175
6.3.3 线程状态的改变	176
本章小结	177
习题	178
第 7 章 网络编程	179
7.1 网络基础知识	179
7.1.1 TCP/IP 协议	179
7.1.2 UDP 协议与 TCP 协议	181
7.1.3 IP 地址与端口号	182
7.2 InetAddress 类与 URL 类	183
7.2.1 InetAddress 类	183

7.2.2 URL 类	184
7.3 UDP 通信编程	186
7.3.1 DatagramPacket 类	187
7.3.2 DatagramSocket 类	187
7.3.3 UDP 通信编程实例	188
7.4 TCP 通信编程	194
7.4.1 ServerSocket 类	194
7.4.2 Socket 类	196
7.4.3 TCP 通信编程实例	196
7.5 综合实例	202
本章小结	216
习题	216
第 8 章 数据库编程	217
8.1 JDBC 技术概述	217
8.2 连接 Access 数据库	218
8.3 连接 MySQL 数据库	219
本章小结	221
习题	222
参考文献	225

第 1 章

Java 概述



本章要点

- Java 发展简史、特点。
- Java 开发与运行环境。
- Java 程序类型、程序结构。
- 简单的 Java 应用程序与 Java 小应用程序的开发步骤。

1.1 Java 发展简史与特点

Java 是 Sun 公司于 1995 年 5 月正式推出的面向对象的程序设计语言,在当今程序设计高级语言已非常丰富的形势下,Java 能够脱颖而出,历经 20 多年仍为最流行的程序设计语言之一,有其必然的历史背景和独树一帜的非凡品质。

1.1.1 Java 发展简史

Java 的诞生与计算机语言细致的改进和不断发展密切相关,且与 C++ 有着千丝万缕的联系,而 C++ 又是从 C 语言派生而来的,因此 Java 继承了这两种语言的大部分特性,如:Java 的语法是从 C 继承的,Java 许多面向对象的特性又受到 C++ 的影响。

20 世纪 90 年代初,Sun 公司资助的 Green 项目以 C++ 为基础,开发用于智能型家用电器控制系统的一种新的程序设计语言(即 Oak 语言),后来 Sun 公司将该语言重新命名为 Java 语言,并在 1995 年 5 月 23 日的 Sun World 会议上正式发布了 Java 技术。这项举措立即在 IT 界引起了轰动,这一天也被 IT 界视为 Java 语言诞生日。随着 Java 的不断完善和适合于网络编程,1996 年 Sun 发布了 JDK 1.0,引起广大厂商的兴趣并购买许可证用于产品的开发,其中包括 IBM、Apple、DEC、Netscape、Oracle、Borland、Microsoft、SGI 等大公司,与此同时,各个软件厂商也都提供了对 Java 的接口支持。Java 经历了 JDK 1.0、JDK 1.1、JDK 1.2 及 JDK 1.3 版本,1998 年 12 月,Sun 公司发布了 Java 2 平台,该平台的发布是 Java 发展史上新的里程碑。Sun 公司将 Java 企业级应用平台作为发展方向,到目前 Java 已有了可扩展的企业级应用 Java 2 平台 J2EE (Java 2 Enterprise Edition)、用于工作站和计算机的 Java 标准平台 J2SE

(Java 2 Standard Edition)和用于嵌入式 Java 消费电子平台 J2ME (Java 2 Micro Edition)三大成员。

Java 的迅猛发展得益于 Internet 的广泛应用,Internet 上的计算机使用了不同的操作系统和 CPU(central processing unit,中央处理器),只要安装了 Java 虚拟机就能够执行相同的 Java 程序。Java 以强大的功能成为当今网络时代的首选编程语言,也大大推动了分布式系统的快速开发和应用,为此,Java 语言拥有“互联网上的世界语”美誉。

1.1.2 Java 特点

Java 是目前使用最广泛的网络编程语言,Sun 公司白皮书将其描述为一种具有简单性、面向对象、可移植性、安全性、稳健性、分布性、多线程、动态性的程序设计语言。

1. 简单性

Java 由 C++ 简化改进而来,但略去其中指针、运算符重载、联合数据类型、类多重继承等难以理解、极少使用的模糊概念和功能,对于熟悉 C、C++ 的程序设计者或 Java 初学者,只需要理解一些基本概念,就可以编写出适合各种情况的应用程序。Java 通过增加垃圾收集功能来实现自动回收内存中的无用信息,从而大大简化了设计者的内存管理工作。同时 Java 解释器、系统模块和运行模块都较小,便于在各种机型上运行,更适合从网上下载资料。

2. 面向对象

Java 是一种完全面向对象的程序设计语言,它提供了简单的类机制和动态的接口(interface)模型,使程序设计者的设计焦点集中于对象及其接口,这里的对象是指应用程序的数据及其操作方法,且 Java 只支持类的单继承,涉及多继承问题是通过接口机制来解决,从而实现了模块化、信息封装和代码的重用,使 Java 面向对象编程变得更加灵活。

3. 可移植性

Java 的基本数据类型长度是固定而独立于平台的,字符串使用标准的 Unicode 字符集进行存储,类库也实现了可移植不同平台的接口。Java 程序编译后产生的中间码是一种与具体机器指令无关的指令集合,这种代码可以在任何一台安装了 Java 虚拟机(JVM)的计算机上正确运行。与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同机器。

4. 安全性

Java 是一种安全的网络编程语言,不支持指针,一切对内存的访问都必须通过对象的实例来实现,这样既能够防止他人使用欺骗手段访问对象的私有成员,也能够避免指针操作中容易产生的错误。此外,Java 的运行环境还具有字节码校验器、运行时内存布局、类装载器和文件访问限制等安全保障机制,以及 Java 虚拟机的“沙箱”运行模式,这些都能有效地防止病毒的侵入和破坏行为的发生。

5. 稳健性

由于 Java 程序在编译和运行时都要对可能出现的问题进行检查,因此在多种情况下都能

稳定执行。Java 有一个专门的指针模型,其作用是防止内存中数据出现被覆盖或毁损的可能,同时,Java 还提供了集成面向对象的异常处理机制,对出现的错误进行控制和处理,以防止系统的崩溃。

6. 分布性

Java 是面向网络的语言,它提供了内容丰富的网络类库,可以处理 HTTP、FTP 等 TCP/IP 协议,它还提供了一个名为 URL 的对象,利用该对象 Java 程序可以很方便地访问网络资源。Java 小程序可以从服务器下载到客户端,将部分计算放在客户端进行,提高系统的执行效率。

7. 多线程

多线程是开发功能强大、复杂程序所必备的手段之一,Java 同样具备多线程机制,通过使用多线程,程序设计者可以分别使用不同的线程完成特定的行为,进而有效提高实时响应能力。Java 的同步机制也保证了对共享数据的共享操作。Java 多线程技术使网上实时交互实现更容易,并为解决网上大数量的客户访问提供了技术基础。

8. 动态性

Java 比 C++ 语言更具动态性,更能适应不断发展变化的运行环境。Java 的类是在运行时动态加载的,因此在类库中,可以自由地加入新的方法和实例变量,但不会影响用户程序的执行,并且 Java 通过接口机制支持多重继承,使之比严格的类继承更具有灵活性和扩展性。

1.2 Java 开发与运行环境

Java 程序的开发通常需要编写源程序、编译生成字节码和运行三个过程,除编写源程序可以使用任何文本编辑器(如操作系统自带的记事本、写字板等)完成外,编译和运行还需要在用户计算机系统中安装所谓的 Java 平台。Java 平台由 Java 虚拟机(Java Virtual Machine,简称 JVM)和 Java 应用程序接口(Application Programming Interface,简称 API)构成,Oracle 公司(注:2009 年收购 Sun 公司)为 Java 平台提供可免费下载的 Java 开发工具集(Java Developers Kits,简称 JDK)。本节将简单介绍 Java 平台、Java 开发工具集(JDK)下载与配置、Eclipse 集成开发环境下载与安装。

1.2.1 Java 平台

Java 程序的开发流程如图 1-1 所示。



图 1-1 Java 程序的开发流程

Java 虽然是遵循“跨平台”思路开发出来的解释型编程语言,但与传统解释型语言不同,Java 源程序首先经过 Java 编译器编译成特定的二进制字节码。Java 字节码是一种与具体机器指令无关的指令集合,不能被计算机直接执行,还需要由 Java 虚拟机中的 Java 解释器执行。

Java 虚拟机是 Java 平台的基础、Java 运行环境的核心,Java 虚拟机中的 Java 解释器(java.exe)负责将 Java 字节码解释成特定的机器码并执行。也就是说,任何安装有 Java 虚拟机的处理器(包括计算机和其他电子设备)都可以安全并且兼容地执行 Java 字节码,而不论最初开发 Java 程序的是何种计算机系统,从而实现了 Java 的跨平台和可移植特性。

Java 平台中的 Java 应用程序接口(API)是大量已编译好的程序代码库,它使程序员可以直接添加现成可定制的功能,以节约编程时间。通常把 Java API 称为类库,它提供了丰富的 Java 资源,许多 Java 平台还补充了扩展类。

Java 源程序、Java 平台和操作系统的关系如图 1-2 所示,从中可看出 Java 平台是 Java 源程序与操作系统之间的接口。

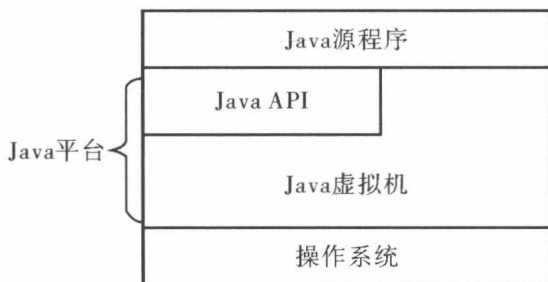


图 1-2 Java 平台与 Java 源程序和操作系统的关系

1.2.2 JDK 下载与安装

开发 Java 程序,必须先安装 Java 开发工具集(JDK)。安装 JDK 后,系统便为 Java 程序提供了开发和运行环境。读者可以很方便地从 Oracle 公司网站(<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)或其他软件站点,下载适合于自己计算机操作系统的 JDK。在此仅介绍在 Windows 7 操作系统下的 JDK 的安装与使用,编写本书时最新的 JDK 版本是 JDK8,下载的 JDK 安装文件名为“jdk-8u131-windows-x64.exe”。

1. 安装 JDK

(1) 双击下载成功的 JDK 安装文件“jdk-8u131-windows-x64.exe”,出现如图 1-3 所示的 JDK 安装向导界面。



图 1-3 JDK 安装向导界面

(2) 单击“下一步”按钮进入 JDK 安装选择界面，如图 1-4 所示。

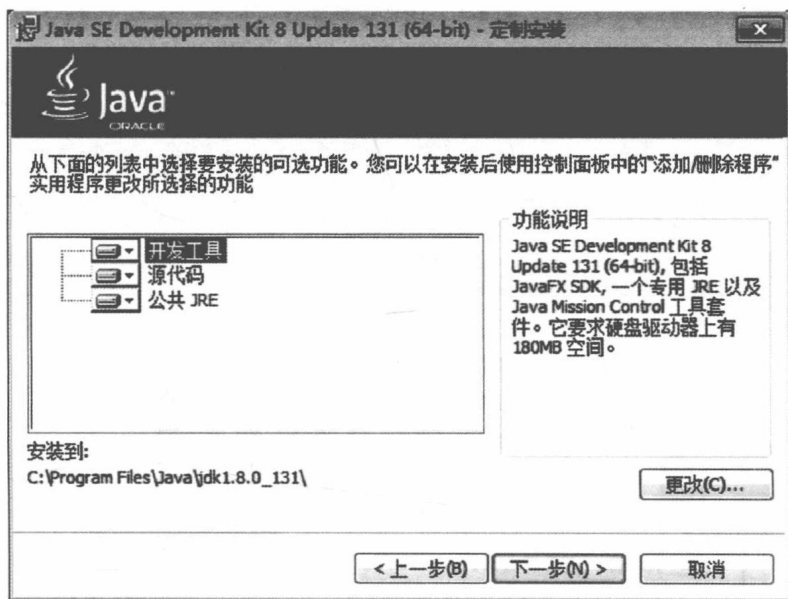


图 1-4 JDK 安装选择界面

(3) 单击“更改”按钮可选择安装目录，在自定义安装程序的功能时，建议选择全部功能。

(4)单击“下一步”按钮进行安装,弹出如图 1-5 所示的 Java 运行环境(Java Runtime Environment, JRE)的安装提示,单击“更改”按钮可更改目标文件夹(如选择“D:\Program Files\Java\jdk_jre”);单击“下一步”按钮出现 JRE 安装过程界面,如图 1-6 所示。

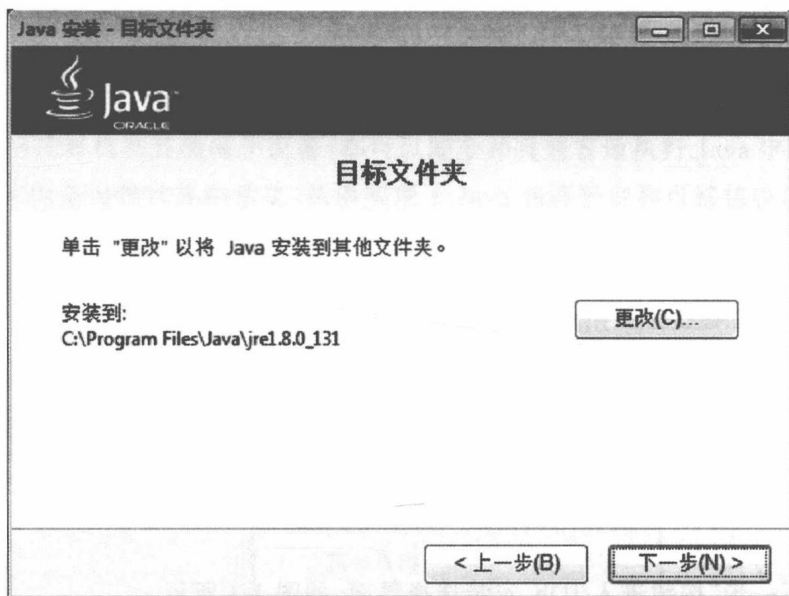


图 1-5 JRE 安装提示

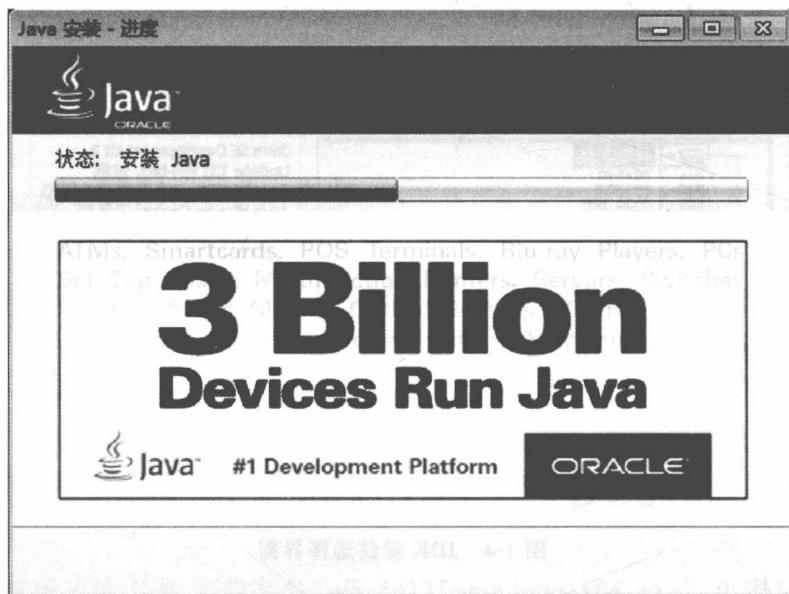


图 1-6 JRE 安装过程界面

(5)最后弹出如图 1-7 所示的安装成功界面,单击“关闭”按钮,即完成 JDK 的安装。

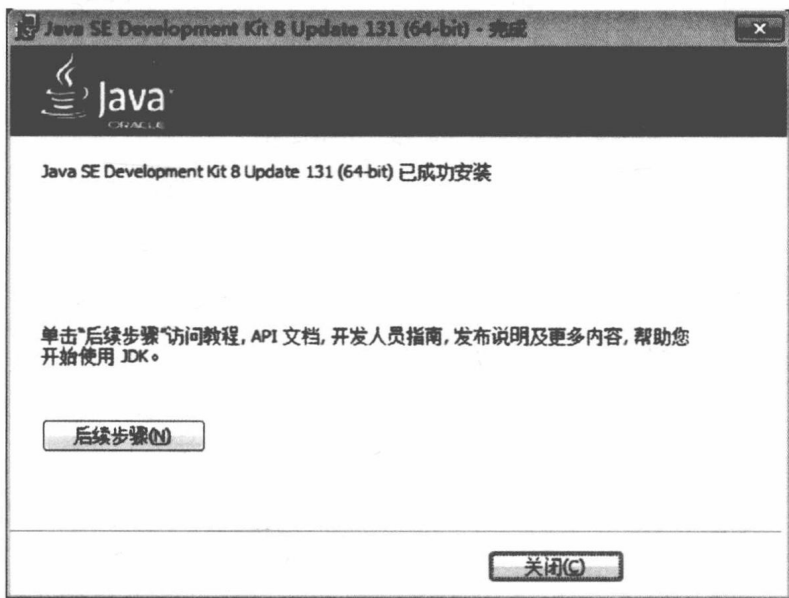


图 1-7 JDK 安装成功界面

2. 设置环境变量

JDK 安装成功后,还需要对操作系统的环境变量进行设置。

(1)在 Windows 操作系统桌面上右击“计算机”图标,在快捷菜单中选择“属性”命令,在如图 1-8 所示“系统”窗口单击左侧面板上的“系统保护”按钮,在随后弹出的“系统属性”对话框中选中“高级”选项,如图 1-9 所示。

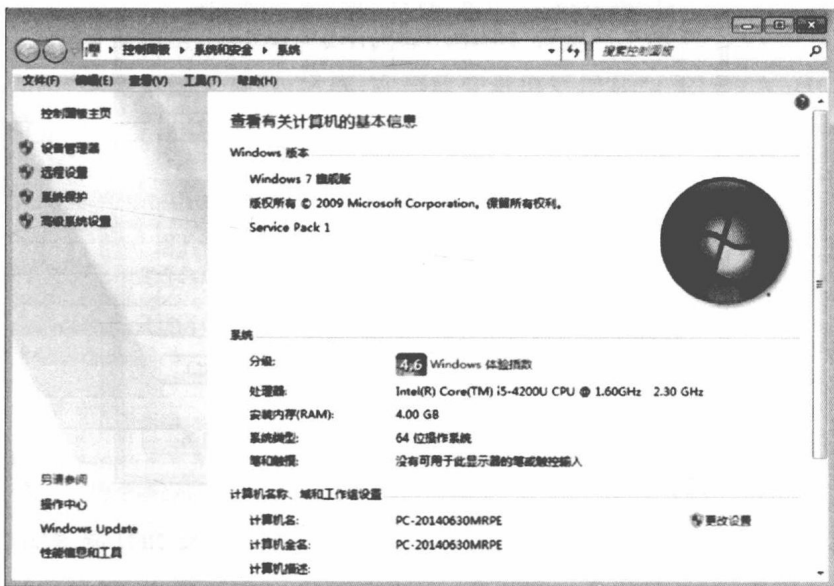


图 1-8 “系统”窗口