

Java程序设计

主编◎杨 健 副主编◎朱云霞 成小惠

- 零基础学Java，结构紧凑、循序渐进，符合语言学习的认知特点
- 问题驱动，在分析、解决问题的过程中提高编程能力
- 例证丰富，方便对抽象概念的感性认识，学用结合



北京邮电大学出版社
www.buptpress.com

Java程序设计

主 编 杨 健
副主编 朱云霞 成小惠



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书是 Java 语言程序设计的一本精编教材, 基于 Java 语言介绍面向对象程序设计的原理与方法。本书采用最新的 Java 技术, 以翔实的例题介绍如何使用 Java 语言进行面向对象的程序设计、GUI 程序设计、文件的输入输出以及线程的程序设计方法。

本书可以作为 Java 语言程序设计课程的通用教材, 适用于各类编程人员、对编程有要求的相关专业的本科生和研究生, 也可作为 Java 技术的自学者或短训班人员使用。

图书在版编目 (CIP) 数据

Java 程序设计 / 杨健主编. -- 北京: 北京邮电大学出版社, 2016. 8

ISBN 978-7-5635-4782-1

I. ①J… II. ①杨… III. ①Java 语言—程序设计—高等学校—教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2016) 第 124870 号

书 名: Java 程序设计

主 编: 杨 健

责任编辑: 王丹丹

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号 (邮编: 100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京通州皇家印刷厂

开 本: 787 mm×1 092 mm 1/16

印 张: 14.75

字 数: 373 千字

印 数: 1—3 000 册

版 次: 2016 年 8 月第 1 版 2016 年 8 月第 1 次印刷

ISBN 978-7-5635-4782-1

定 价: 30.00 元

• 如有印装质量问题, 请与北京邮电大学出版社发行部联系 •

前 言

Java 技术 1995 年 5 月由 Sun 公司推出,是 Java 程序设计语言和 Java 平台的总称,用 Java 实现的浏览器(支持 Java Applet)显示了 Java 的魅力,如跨平台、动态的 Web、Internet 计算等。Java 由 Java 虚拟机(Java Virtual Machine, JVM)和 Java 应用编程接口(Application Programming Interface, API)构成。Java 应用编程接口为 Java 应用提供了一个独立于操作系统的标准接口,在硬件或操作系统平台上安装 Java 环境后,应用程序就可跨平台运行。现在,越来越多的领域都采用 Java 作为开发语言,Java 语言也成为软件开发人员理想的工具。

本书是 Java 语言程序设计的一本精编教材,基于 Java 语言介绍面向对象程序设计的原理与方法。本书采用最新的 Java 技术,以大量翔实的例题介绍如何使用 Java 语言进行面向对象的程序设计、GUI 程序设计、文件的输入输出以及线程的程序设计方法。

本书共 10 章。第 1 章介绍了 Java 语言的发展过程、Java 语言的特点以及 Java 平台的工作原理,详细地讲述了安装和配置 Java 开发工具的步骤;第 2 章介绍了 Java 语言的基本语法;第 3 章结合面向对象的程序设计思想,通过对类和对象的详细介绍,来阐述用 Java 语言实现面向对象思想中的抽象性和封装性,重点讲述类与对象的概念、类的定义、对象的使用和包的使用;第 4 章介绍类的继承机制;第 5 章介绍接口和标注;第 6 章介绍异常;第 7 章介绍文件的管理与输入输出;第 8 章介绍线程;第 9 章介绍图形用户界面开发;第 10 章介绍 Java 中的实用包。

全书由课程组集体编写,在编写过程中得到南京邮电大学领导的支持,还有课程组杨健、刘尚东、成小惠、朱云霞、周莉、朱艳梅、肖欣欣、陈兴国、徐力杰的协助和支持,在此一并感谢。教材编写时间紧迫,难免有疏漏,欢迎各位读者对本书提出批评和修改建议。我们将非常感激,并在再版时予以考虑。

目 录

第 1 章 Java 入门	1
1.1 Java 语言概述	1
1.1.1 Java 的发展	1
1.1.2 Java 的特点	2
1.2 Java 平台工作原理	3
1.2.1 JVM 介绍	3
1.2.2 Java 运行流程	4
1.3 Java 开发环境	4
1.3.1 JDK 的安装和使用	5
1.3.2 Eclipse 的安装和使用	9
1.3.3 JCreator 的安装和使用	11
1.4 实现第一个 Java 程序	13
1.4.1 简单的 Application 程序	14
1.4.2 简单的 Applet 小应用程序	15
1.4.3 Java 的注释	17
小结	17
习题	18
第 2 章 Java 基本语法	19
2.1 标识符与关键字	19
2.1.1 标识符	19
2.1.2 关键字	19
2.2 数据与数据类型	20
2.2.1 数据的类型	20
2.2.2 基本数据类型	20
2.2.3 复合数据类型	22
2.3 常量与变量	22
2.3.1 常量	22
2.3.2 变量	23
2.4 基本输入与输出	26
2.4.1 数据的输入	26

2.4.2 数据的输出	30
2.5 运算符与表达式	30
2.5.1 运算符介绍	30
2.5.2 运算符的优先级	35
2.5.3 表达式	35
2.6 程序流程控制	36
2.6.1 语句与程序流程	36
2.6.2 顺序结构	37
2.6.3 选择结构	38
2.6.4 循环结构	45
2.6.5 中断流程控制	50
2.7 数组	52
2.7.1 一维数组	53
2.7.2 二维数组	55
2.7.3 数组的操作	58
2.8 方法	61
2.8.1 方法的定义	62
2.8.2 方法的调用	62
2.8.3 参数传递的方式	63
2.8.4 方法的重载	64
2.8.5 嵌套与递归	65
小结	66
习题	66
第3章 类与对象	68
3.1 面向对象程序设计概述	68
3.1.1 面向对象的基本思想	68
3.1.2 类和对象	69
3.1.3 Java 的面向对象技术	69
3.2 类的创建	70
3.2.1 类的定义	70
3.2.2 成员变量	70
3.2.3 成员方法	71
3.2.4 类定义示例	71
3.3 对象的创建和使用	73
3.3.1 创建对象	73
3.3.2 构造方法与对象初始化	73
3.3.3 对象的使用	76

3.3.4	this 关键字	77
3.3.5	finalize 方法与对象的销毁	78
3.4	类的封装	79
3.4.1	封装的目的	79
3.4.2	类的访问控制	80
3.4.3	成员的访问控制	82
3.4.4	类成员(静态成员)	84
3.5	类的导入与包	86
3.5.1	包的概念	86
3.5.2	package 语句	87
3.5.3	import 语句	87
3.6	综合示例	87
小结		96
习题		96
第4章	继承	97
4.1	类的抽象和扩展	97
4.2	继承的定义	98
4.2.1	语法格式	98
4.2.2	构造器与实例化	98
4.2.3	Super 的使用	101
4.2.4	继承关系下的作用域	102
4.3	方法的覆盖	102
4.4	类型转换	105
4.5	根父类:Object 类	105
4.5.1	认识 Object 类	105
4.5.2	equals()方法使用与继承	106
4.5.3	hashCode()方法使用与继承	107
4.5.4	toString()方法使用与继承	107
4.6	枚举类型	108
4.6.1	枚举类型的定义	108
4.6.2	枚举类型与类	109
4.6.3	Enum 类	109
4.7	继承的设计与应用	110
4.7.1	类的层次结构设计	110
4.7.2	使用继承设计用户界面	111
小结		111
习题		111

第 5 章 抽象、接口与标注	114
5.1 抽象类	114
5.1.1 概述	114
5.1.2 抽象类的语法	115
5.2 接口	117
5.2.1 概述	117
5.2.2 接口语法	118
5.2.3 接口的扩展	119
5.2.4 起标签作用的接口	120
5.2.5 如何使用接口	120
5.3 标注	121
5.3.1 标注(Annotations)概述	121
5.3.2 标注类型中用到其他标注类型	122
5.3.3 没有元素的标注类型	123
5.3.4 标注元素的初始化	123
5.3.5 限制标注的使用	123
5.3.6 标注类型的使用限定	124
5.3.7 标注类型的使用原则	125
小结	125
习题	125
第 6 章 异常与断言	126
6.1 异常	126
6.1.1 基本概念	126
6.1.2 异常语法	127
6.1.3 finally 块	127
6.1.4 throw 语句	128
6.1.5 throws 语句	128
6.1.6 throws 语句和方法重写(Override)	131
6.1.7 异常链	132
6.1.8 异常使用原则	132
6.2 断言(Assertion)	133
6.2.1 断言(Assertion)概述	133
6.2.2 断言语法	133
6.2.3 断言使用原则	133
小结	134
习题	134

第 7 章 文件管理与输入/输出	135
7.1 File 类	135
7.2 流	139
7.2.1 读写字节	139
7.2.2 流过滤	142
7.2.3 标准输入与输出流	143
7.3 二进制数据读写	144
7.3.1 DataOutputStream 类与 DataInputStream 类	145
7.3.2 RandomAccessFile 类	147
7.4 文本数据读写	148
7.4.1 写文本数据	149
7.4.2 读文本数据	151
7.5 ZIP 文件读写	153
7.5.1 写 ZIP 文件	154
7.5.2 读 ZIP 文件	155
7.6 对象序列化	156
小结	158
习题	158
第 8 章 线程	160
8.1 理解线程	160
8.1.1 进程的概念	160
8.1.2 线程的概念	160
8.1.3 线程的生命周期	161
8.2 线程类设计	161
8.2.1 线程 API 类图	161
8.2.2 线程类 Thread 的构造方法	162
8.3 线程实现	162
8.3.1 继承实现	162
8.3.2 接口实现	163
8.4 线程控制	163
8.4.1 监控线程状态	163
8.4.2 线程睡眠	164
8.4.3 中断线程	164
8.4.4 阻塞线程	164
8.4.5 线程等待和唤醒	165
8.4.6 线程终止	165

8.5 线程属性	167
8.5.1 优先级属性	167
8.5.2 守护线程	168
8.6 线程同步	168
8.6.1 Synchronized 同步方法和同步代码块	168
8.6.2 同步变量 volatile 关键字	169
8.6.3 线程锁 Lock 接口及 ReentrantLock 类	171
8.6.4 死锁	172
8.6.5 测试锁	173
8.6.6 读写锁 ReadWriteLock	174
小结	176
习题	176
第9章 图形用户界面设计	177
9.1 AWT 和 Swing 概述	177
9.1.1 AWT 概述	177
9.1.2 Swing 概述	178
9.1.3 AWT 与 Swing 的区别	179
9.2 事件处理	179
9.2.1 事件	179
9.2.2 事件处理模型	179
9.2.3 事件处理的实现	180
9.3 容器	183
9.3.1 框架 JFrame	183
9.3.2 面板 JPanel	184
9.3.3 对话框 JDialog	186
9.4 文本组件	191
9.4.1 标签 JLabel	191
9.4.2 文本框 JTextField	192
9.4.3 密码框 JPasswordField	192
9.4.4 文本域 JTextArea	192
9.4.5 富文本 JTextPane	193
9.4.6 文本组件的事件处理	193
9.5 选择组件	193
9.5.1 按钮 JButton	193
9.5.2 复选框 JCheckBox	194
9.5.3 单选框 JRadioButton 和单项按钮组 ButtonGroup	194
9.5.4 组合框 JComboBox	195

9.5.5 列表 JList	196
9.6 菜单	196
9.6.1 顶层菜单	197
9.6.2 弹出式菜单	198
9.6.3 菜单项	200
9.7 布局管理器	201
9.7.1 BorderLayout	201
9.7.2 FlowLayout	202
9.7.3 GridLayout	203
9.7.4 BoxLayout	203
小结	205
习题	205
第 10 章 Java 实用包	207
10.1 常用数学函数	207
10.2 字符串处理	209
10.2.1 String 类	209
10.2.2 StringBuffer 类	213
10.3 数组处理	215
10.3.1 Arrays 类	215
10.3.2 ArrayList 类	218
小结	222
习题	222
参考文献	223

第 1 章 Java 入门

Java 语言由 Sun 公司于 1995 年 5 月 23 日正式推出,是一个面向对象、基于网络及支持多媒体的程序设计语言。Java 技术具有卓越的通用性、高效性、平台移植性和安全性,它广泛应用在桌面系统、Web 系统、分布式系统及嵌入式系统,同时拥有全球最大的开发者专业社群。

Java 易学易用,功能强大,并提供了丰富的类库,开发人员可以方便地构建项目和开发大型系统。

本章学习目标:

1. 了解 Java 的发展过程和特点。
2. 学会 JDK 的安装及环境配置。
3. 掌握 Java 集成开发环境的安装和使用。
4. 了解 Java 程序的基本结构及注释。
5. 掌握 Java 程序的开发过程。

1.1 Java 语言概述

1.1.1 Java 的发展

1990 年 Sun Microsystems 公司为发展消费类电子产品进行了一个名为 Green 的项目。这个项目的负责人是 James Gosling,项目组开发第一个版本花了 18 个月时间,该语言最初名称为“Oak”。他们最初的目的只是为了开发一种独立于平台的软件技术。到 1994 年,Green 小组发现他们的新型编程语言“Oak”比较适合 Internet 程序的编写,于是他们对“Oak”进行改进和完善,并获得了巨大的成功。他们用“Oak”语言开发了一个实时性较高、可靠安全、有交互功能的新型 Web 浏览器,名为 HotJava。后来发现“Oak”已经是 Sun 公司另外一种语言的注册商标,1995 年 1 月,“Oak”更名为 Java。这个名字的产生,来自于印度尼西亚一个盛产咖啡的岛屿,中文名为爪哇,寓意是为世人送上一杯热气腾腾的咖啡。

Java 语言发展非常迅速,1995 年 Sun 公司发布了 Java 语言的 Alpha1.0 版本,1996 年 1 月发布了 Java 语言的第一个开发包 JDK1.0,1997 年 2 月发布了 Java 语言的开发包 JDK1.1,从而奠定了 Java 语言在计算机语言中的地位。1998 年 12 月,Sun 公司发布了 JDK1.2,也是 Java 历史上最重要的一个 JDK 版本。这个版本标志着 Java 进入了 Java2 时代。2009 年 4 月,Oracle 公司通过收购 Sun 公司获得 Java 版权。2011 年 7 月 28 日,甲骨文发布 Java 7.0 的正式版。2014 年 3 月 19 日,甲骨文公司发布 Java 8.0 的正式版,这也是目前的最新版本。

20 多年来,Java 语言的发展获得了巨大的成功,就像爪哇岛的咖啡一样誉满全球,已成为当今最流行的程序设计语言之一。

1.1.2 Java 的特点

Java 是一种简单、面向对象、分布式、解释型、健壮、安全、与平台无关、可移植、性能高效、多线程的动态语言。

1. 简单性

Java 语言的语法与 C 语言和 C++ 语言很接近,大多数程序员很容易学习和使用。Java 摒弃了 C++ 中很少使用和难以理解的一些特性,如操作符重载、类的多继承、自动的强制类型转换。特别是 Java 语言不使用指针,并提供了自动的垃圾收集,使得程序员不必为内存管理而担忧。

2. 面向对象

Java 语言是一个完全面向对象的程序设计语言。它吸收了 C++ 面向对象的概念,将数据封装在类中,实现了程序的简洁性和便于维护性。Java 语言的程序设计集中在对象和接口上,并提供了简单的类机制和动态接口模型。为了简单起见,只支持类之间的单继承,但支持接口之间的多继承,并支持类与接口之间的实现机制。通过继承机制,子类可以使用父类所定义的方法,以便实现程序的重复使用。

3. 分布式

Java 语言支持 Internet 应用的开发,在基本的 Java 应用编程接口中有一个网络应用编程接口(java.net),它提供了用于网络应用编程的类库,包括 URL、URLConnection、Socket、ServerSocket 等。Java 的 RMI(远程方法激活)机制也是开发分布式应用的重要手段。

4. 解释性

Java 程序在 Java 平台上被编译为字节码格式,然后可以在安装了 Java 平台的任何系统中运行。在运行时,Java 平台中的 Java 解释器对这些字节码进行解释执行,执行过程中需要的类在连接阶段被载入到运行环境中。

5. 健壮性

用 Java 编写的程序有多方面的可靠性和稳定性,程序在编译和运行时要对可能出现的问题进行检查,并消除了有出错倾向的状态。Java 的强类型机制、异常处理、垃圾的自动收集等是 Java 程序健壮性的重要保证。Java 通过集成面向对象的异常处理机制,在编译时提示可能出现但未被处理的异常,以防止系统崩溃。Java 的安全检查机制使得 Java 更具健壮性。

6. 安全性

Java 不支持指针,杜绝了内存的非法访问。Java 通过对象的实例来实现对内存的访问,这样可以防止他人使用欺骗手段访问对象的私有成员,也能避免指针在操作中易产生的错误。

Java 编写的程序通常应用在网络环境中,Java 必须提供足够的安全保障,并且能够防止恶意代码的攻击。Java 在运行应用程序时,严格检查其访问数据的权限,例如不允许网络上的应用程序修改本地的数据。下载到用户计算机中的字节码在其执行前要进行核实,一旦字

节代码被核实,便由 Java 解释器来执行,该解释器通过阻止对内存的直接访问来进一步保证 Java 的安全性。

7. 与平台无关

Java 是与平台无关的语言,用 Java 编写的应用程序不用修改就可以运行在不同的软硬件平台上。Java 编译器能够产生一种与计算机体系结构无关的字节码(Byte Code),只要安装了 Java 虚拟机(Java Virtual Machine,JVM),Java 就可以在相应的处理机上执行。

JVM 是一种抽象机器,它运行在具体操作系统之上,本身具有一套虚拟机的机器指令,并有自己的栈、寄存器组等。但 JVM 通常是在软件上而不是在硬件上实现。

8. 可移植

Java 具有良好的可移植性,主要得益于它与平台无关的特性。另外,Java 的类库中也实现了与平台无关的接口,使得这些类库也能移植。

Java 解释产生的目标代码是针对一种并不存在的 CPU,即 Java 虚拟机,JVM 避免了不同 CPU 之间的差别,使编译过的字节码能运行于任何安装了 JVM 的机器上。

9. 高性能

与那些解释型的高级脚本语言相比,Java 的确是高性能的。Java 系统提供了 JIT(Just In Time)编译器,JIT 能够产生编译好的本地机器代码,以提高 Java 代码的执行速度。Java 的运行速度随着 JIT 编译器技术的发展,已经具有与 C++ 同样,甚至有些情况下更好的运行性能。

10. 多线程

在 Java 语言中,多线程是非常重要的组成部分。Java 语言支持多个线程机制使应用程序能够并发执行,而且提供的同步机制保证了对数据的共享操作。通过使用多线程,程序员可以分别利用不同的线程来完成特定的行为,而不需要采用全局的事件循环机制,这样就很容易实现网络的实时交互行为,从而为解决网上大量用户的访问提供了技术基础。

11. 动态性

Java 语言的设计目标之一是适应动态变化的环境。Java 程序需要的类能够动态地被载入到运行环境中,也可以通过网络来载入所需要的类,这也有利于软件的升级。另外,Java 中的类有一个运行时刻的表示,能进行运行时刻的类型检查。

1.2 Java 平台工作原理

1.2.1 JVM 介绍

JVM 全称 Java Virtual Machine,是一种利用软件方法来实现硬件功能的虚拟计算机,JVM 的任务是执行 Java 程序。JVM 是一种用于计算设备的规范,可以用软件实现,也可以用硬件实现,目前大多数用软件实现。JVM 是一个可以执行 Java 字节码的操作平台,只要根据 JVM 规格描述将解释器移植到特定的计算机上,就能保证经过编译的任何 Java 字节代码能够在该系统上运行。

Java 语言的一个非常重要的特点就是与平台的无关性,而使用 JVM 是实现这一特点的关键。一般的高级语言如果要在不同的平台上运行至少需要编译成不同的目标代码。而引入 JVM 后,Java 语言在不同平台上运行时不需要重新编译。Java 语言使用 JVM,屏蔽了与具体平台相关的信息,Java 语言的编译程序可以在多种平台上不加修改地运行。Java 虚拟机在执行时,把字节码解释成具体平台上的机器指令来执行。

对 Java 程序进行解释也有助于它的安全性。因为每个 Java 程序的运行都处于 Java 虚拟机的控制之下,Java 虚拟机可以包含这个程序并且能阻止它在系统之外产生副作用。

1.2.2 Java 运行流程

计算机高级语言类型主要有编译型和解释型两种,Java 程序却比较特殊,它是两种类型的结合,先经过编译,再经过解释才能执行。

如图 1-1 所示,用 Java 编写的代码为源程序,是以 .java 为扩展名的文件,经过 Java 编译系统编译后生成扩展名为 .class 的字节码文件,然后由 JVM 解释为可执行的字节码文件。字节码文件不能直接在操作系统上运行,而只能通过虚拟机解释执行。利用 Java 虚拟机可以把 Java 字节码程序跟具体的操作系统及硬件分隔开来,只要在各种平台上都实现了 Java 虚拟机,任何 Java 程序都可以在该系统上运行,实现了“一次编程,到处运行”的目标。

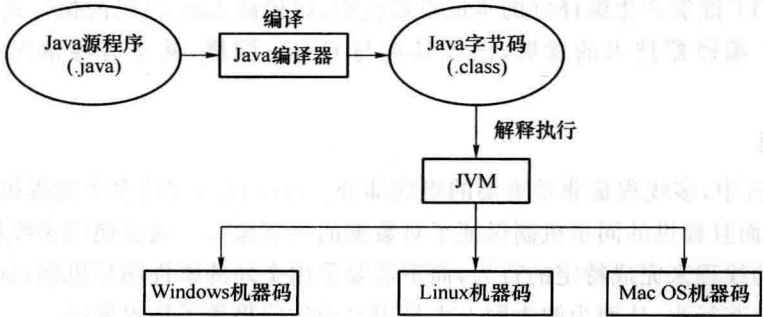


图 1-1 Java 运行原理

当 Java 程序用 Java 编译器编译成字节码后,便可运行在任何含有 Java 虚拟机的平台上,无论是 Windows、UNIX/Linux 或 Mac OS。Java 这种跨平台的特点,也使 Java 得到了快速发展。

1.3 Java 开发环境

Java 开发环境主要有两种:一种是使用基础开发工具;另一种是使用集成开发环境(Integrate Development Environment, IDE)。常用的基础开发工具是由 Oracle 公司提供的免费开发工具(Java Development Kits, JDK),它是以 DOS 命令行的方式使用的。

Java 的集成开发环境为程序员提供了更方便的交互开发平台,它将 Java 程序编辑、编译、运行与调试以及项目管理等一系列的工程集成到一起,并且是基于图形用户界面的。当设计比较复杂的项目时,为了提高开发效率和实现对项目的管理,应该使用 Java 的集成开发环境。Java 的 IDE 有多种,如 Jcreator、Eclipse、MyEclipse、NetBean、IDEA 等。

1.3.1 JDK 的安装和使用

1. JDK 的下载和安装

JDK 主要分为 3 种版本:Java SE、Java EE、Java ME。

Java SE 称为 Java 的标准版或标准平台,Java SE 提供了标准的 JDK 开发平台,使用该平台可以开发 Java 桌面应用系统和低端服务器应用程序,也可以开发 Java Applet 程序。

Java EE 称为 Java 企业版或企业平台,使用 Java EE 可以构建企业级的服务应用,Java EE 平台包含了 Java SE 平台,并且增加了类库,便于支持目录管理、交易管理等功能。

Java ME 称为 Java 的微型版或小型平台,Java ME 是一种很小的 Java 运行环境,用于嵌入式的消费产品中,如移动电话、掌上电脑或其他无线设备等。

登录 Oracle 公司网站 <http://www.oracle.com>,将看到有关 J2SE、J2EE、J2ME 的信息。三种版本的 Java 运行平台,都包含了相应的 Java 虚拟机。Java 虚拟机负责将字节码文件加载到内存,然后以解释的方式执行字节码文件。

适应于普通 PC 应用的 Java 开发平台为 Java SE,其全称是“Java Platform, Standard Edition”,它是 Java 开发的标准版本。我们可以很方便地从 Java 的官方网站免费下载。JDK 有多个版本,高版本可以对低版本实现向下兼容。

另外,Oracle 公司提供了 Java 的帮助文档也可以在 <http://www.oracle.com> 网站下载。

登录 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 网站,下载适合自己计算机操作系统的 JDK,在 Windows 32 操作系统下,下载最新的 JDK 开发工具“jdk-8u77-windows-i586.exe”;而在 Windows 64 操作系统下,下载最新的 JDK 开发工具“jdk-8u77-windows-x64.exe”。如果在 Windows 32 操作系统下安装,则在下载完成后直接运行“jdk-8u77-windows-i586.exe”,按照安装向导进行安装。如图 1-2 所示,为 JDK 安装向导的第一个界面,单击【下一步】按钮,进入配置对话框,如图 1-3 所示。在配置对话框中主要有开发工具、源代码、公共 JRE 几个可选功能,建议初学者把这样的功能都安装上,以避免使用时出现麻烦。对于安装目录,单击【更改】按钮可以进行更改,也可以用默认目录“C:\Program Files (x86)\Java\jdk1.8.0_65”安装。

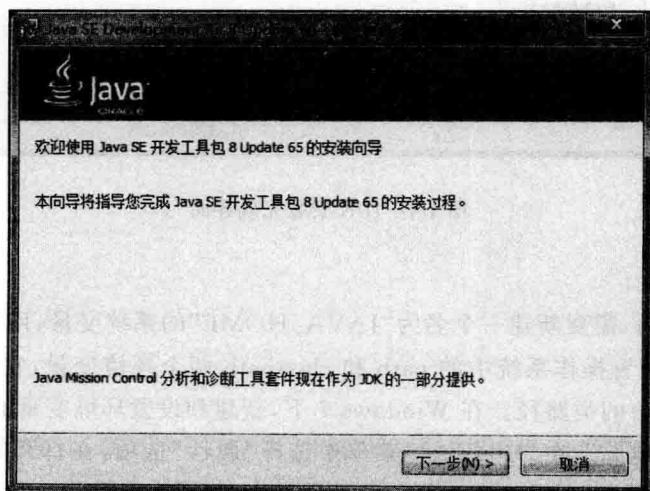


图 1-2 JDK 安装向导界面

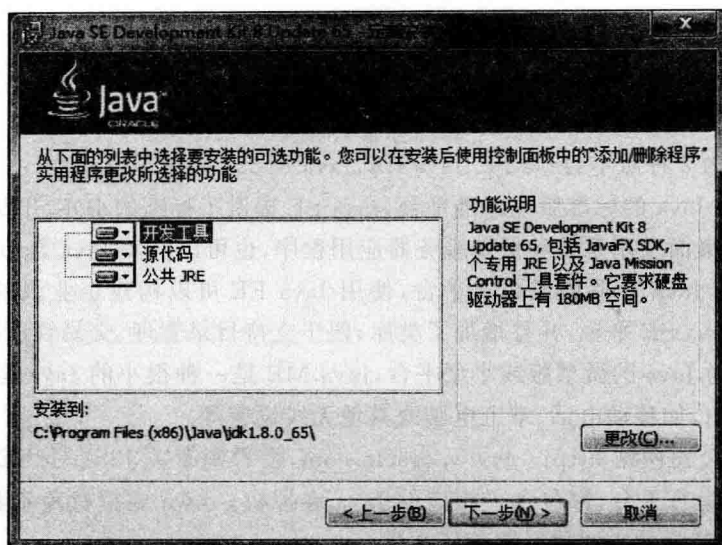


图 1-3 选择安装目录

在图 1-3 所示的界面中,单击【下一步】按钮,Java 自动进行安装,如图 1-4 所示,JDK 安装完成,单击【关闭】按钮。

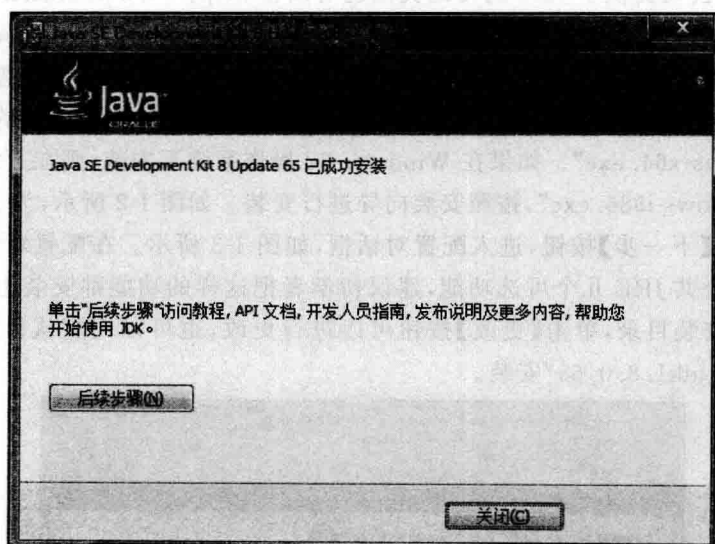


图 1-4 JDK 安装完成界面

2. 设置 JDK

JDK 安装完成后,需要新建一个名为“JAVA_HOME”的系统变量,用于指定 JDK 的安装路径。另外还必须配置操作系统中的 path 和 classpath 两个环境变量,它们分别指定了 Java 工具包的路径和 Java 的类路径。在 Windows 7 下,新建和设置环境变量的过程如下。

(1) 右击“我的电脑”,在弹出的快捷菜单中选择“属性”选项,在打开的窗口中单击“高级系统设置”,在“系统属性”对话框中选择“高级”选项卡,如图 1-5 所示。单击【环境变量】按钮,弹出“环境变量”对话框,如图 1-6 所示。