

Java 程序设计 (第2版)

主编 林邓伟 孟军霞

Java 程序设计

(第2版)

主 编 林邓伟 孟军霞

副主编 崔 艳 程跃华 赵小丽 李亚波 李东亮

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

版权专有 侵权必究

图书在版编目 (CIP) 数据

Java 程序设计/林邓伟, 孟军霞主编. —2 版. —北京: 北京理工大学出版社,
2012. 11

ISBN 978 - 7 - 5640 - 7006 - 9

I. ①J… II. ①林…②孟… III. ① JAVA 语言 - 程序设计 - 教材 IV.
①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 266296 号

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010)68914775(办公室) 68944990(批销中心) 68911084(读者服务部)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 天津紫阳印刷有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 18

字 数 / 417 千字

版 次 / 2012 年 11 月第 2 版 2012 年 11 月第 1 次印刷

印 数 / 1 ~ 1500 册

定 价 / 52.00 元

责任编辑 / 钟 博

责任校对 / 杨 露

责任印制 / 王美丽

图书出现印装质量问题, 本社负责调换

Java 语言自 1995 年出现以来,就以迅猛的发展速度成为计算机主流编程语言。跨平台、完全面向对象、既适于单机编程也适于 Internet 编程等特点,给 Java 语言注入了强大的生命力。如今,Java 在商业、通信和工业等很多领域开始大量应用,Java 技术平台已经成为世界上最卓越的企业应用和移动应用开发平台之一。

本书以项目模块为明线,课程知识为暗线,将理论知识与实践知识整合,将知识融入项目之中。在编写风格上,本书不追求对技术的全面介绍,而是根据项目需要,通过实例引入、知识讲解、分析提高的递进方式适当讲解,并通过项目拓展的方式启发读者对相关知识的学习,符合人们对事物的认识规律。本书第 1 版自 2010 年出版以来,得到了许多读者的认同。

本书内容包括 Java 语言入门、Java 语言基础、Java 面向对象编程、异常处理、Applet 程序设计、图形用户界面、输入与输出、多线程、Java 多媒体技术、Java 数据库编程和 Java 网络编程共 11 个一级模块,每个一级模块由项目概述、项目目的、项目支撑知识、项目实施、项目总结、扩展演练组成。每个项目知识通过大量的例题详细讲解项目中的知识点,便于读者循序渐进地掌握 Java 程序开发的方法;知识点与例题相得益彰,使得学习更加有效。每个项目实施部分通过结合经典的实际项目来讲述面向对象和 Java 编程的基本思想、方法和技术,使读者通过项目的开发进一步加深对相关 Java 知识的理解并掌握实际编程技能。

本书内容以最新的 Java 技术版本为背景,紧跟 Java 技术发展潮流。本书层次分明,对基本知识、技术论述详尽,又注重深度与广度的拓展,可按需进行取舍,以适应个性化教育的需

要。本书既可作为高等院校计算机及相关专业的教材，也适合具有一般计算机基础的读者自学使用，还可作为相关培训的教材。

参与本书编写的教师均有丰富的 Java 应用开发与 Java 课程教学的经验。其中，项目 1、11 由焦作大学林邓伟执笔，项目 3、4 由焦作大学孟军霞执笔，项目 5、6 由焦作大学程跃华执笔，项目 7、9 由焦作大学崔艳执笔，项目 2、8 由赵小丽及吉林农业大学李亚波执笔，项目 10 及附录由焦作大学李东亮执笔。全书由林邓伟负责规划与统稿。

由于学识和时间的限制，书中的缺点和欠妥之处在所难免，衷心希望得到专家、读者的批评指正。

编 者
于焦作大学

目录

项目 1	Java 语言入门	1
1.1	项目概述	1
1.2	项目目的	1
1.3	项目支撑知识	2
1.3.1	项目开发背景知识 1——Java 的发展历史	2
1.3.2	项目开发背景知识 2——Java 语言特性	3
1.3.3	项目开发背景知识 3——Java 运行机制	4
1.3.4	项目开发背景知识 4——Java 开发环境	6
1.4	项目实施	7
1.4.1	Java 开发环境的搭建	7
1.4.2	一个简单的 Java 应用程序	11
1.5	项目总结	13
1.6	扩展演练	13
项目 2	Java 语言基础	14
2.1	项目概述	14
2.2	项目目的	14
2.3	项目支撑知识	14
2.3.1	项目开发背景知识 1——Java 基本符号	14
2.3.2	项目开发背景知识 2——基本数据类型	16
2.3.3	项目开发背景知识 3——常量和变量	20
2.3.4	项目开发背景知识 4——运算符和表达式	22
2.3.5	项目开发背景知识 5——流程控制语句	25
2.4	项目实施	37
2.5	项目总结	40
2.6	扩展演练	41
项目 3	Java 面向对象编程	42
3.1	项目概述	42
3.2	项目目的	42
3.3	项目支撑知识	42
3.3.1	项目开发背景知识 1——Java 的面向对象技术	42
3.3.2	项目开发背景知识 2——类和对象	44
3.3.3	项目开发背景知识 3——继承与多态	56
3.3.4	项目开发背景知识 4——类的封装	66
3.3.5	项目开发背景知识 5——抽象类和接口	73
3.3.6	项目开发背景知识 6——系统常用类	77
3.3.7	项目开发背景知识 7——数组	86
3.4	项目实施	89
3.5	项目总结	102

3.6 扩展演练	102
----------------	-----

项目4 异常处理

4.1 项目概述	103
4.2 项目目的	103
4.3 项目支撑知识	103
4.3.1 项目开发背景知识1——异常处理的概念	103
4.3.2 项目开发背景知识2——异常捕获与抛出	105
4.3.3 项目开发背景知识3——自定义异常类	113
4.4 项目实施	115
4.5 项目总结	117
4.6 扩展演练	117

项目5 Applet 程序设计

5.1 项目概述	118
5.2 项目目的	118
5.3 项目支撑知识	118
5.3.1 项目开发背景知识1——Applet 的创建与运行	118
5.3.2 项目开发背景知识2——Applet 的生命周期	121
5.3.3 项目开发背景知识3——Applet 类的显示方法	122
5.3.4 项目开发背景知识4——Applet 的 HTML 标记	125
5.4 项目实施	127
5.5 项目总结	132
5.6 扩展演练	132

项目6 图形用户界面

6.1 项目概述	133
6.2 项目目的	133
6.3 项目支撑知识	133
6.3.1 项目开发背景知识1——AWT 简介	133
6.3.2 项目开发背景知识2——布局管理器	136
6.3.3 项目开发背景知识3——事件处理	141
6.3.4 项目开发背景知识4——AWT 组件	147
6.4 项目实施	154
6.5 项目总结	170
6.6 扩展演练	170

项目7 输入与输出

7.1 项目概述	171
7.2 项目目的	171
7.3 项目支撑知识	171

7.3.1 项目开发背景知识 1——输入/输出流	171
7.3.2 项目开发背景知识 2——文件	172
7.3.3 项目开发背景知识 3——字节流	174
7.3.4 项目开发背景知识 4——字符流	182
7.3.5 项目开发背景知识 5——标准输入/输出流	187
7.4 项目实施	189
7.5 项目总结	196
7.6 扩展演练	196

项目 8 多线程

8.1 项目概述	197
8.2 项目目的	197
8.3 项目支撑知识	197
8.3.1 项目开发背景知识 1——线程概述	197
8.3.2 项目开发背景知识 2——建立线程	201
8.3.3 项目开发背景知识 3——线程同步	204
8.4 项目实施	205
8.5 项目总结	207
8.6 扩展演练	207

项目 9 Java 多媒体技术

9.1 项目概述	208
9.2 项目目的	208
9.3 项目支撑知识	208
9.3.1 项目开发背景知识 1——颜色	208
9.3.2 项目开发背景知识 2——文本和字体	209
9.3.3 项目开发背景知识 3——绘制图形	212
9.3.4 项目开发背景知识 4——显示图像	220
9.3.5 项目开发背景知识 5——播放声音	223
9.4 项目实施	225
9.5 项目总结	230
9.6 扩展演练	230

项目 10 Java 数据库编程

10.1 项目概述	231
10.2 项目目的	231
10.3 项目支撑知识	231
10.3.1 项目开发背景知识 1——JDBC 基础	231
10.3.2 项目开发背景知识 2——JDBC 编程接口	233
10.4 项目实施	235
10.4.1 配置 ODBC 数据源	235
10.4.2 程序实现	239

10.5	项目总结与提高	244
10.6	扩展演练	245

项目 11 Java 网络编程..... 246

11.1	项目概述	246
11.2	项目目的	246
11.3	项目支撑知识	246
11.3.1	项目开发背景知识 1——网络编程概述	246
11.3.2	项目开发背景知识 2——Java 与 Socket 编程	248
11.4	项目实施	252
11.4.1	服务器程序的实现	252
11.4.2	客户端程序的实现	256
11.5	项目总结与提高	260
11.6	扩展演练	265

附录 1 JCreator 的安装与使用

附录 2 Java API 文档

附录 3 JavaDoc 文档

Java 语言入门

在 Windows 系统下编写的程序能够不做修改就直接拿到 UNIX 系统上运行吗？显然是不可以的，因为程序最终必须转换成为计算机硬件的机器指令来执行，专门为某种计算机硬件和操作系统编写的程序是不能直接放到另外的计算机硬件上执行的，至少要做移植工作。要想让程序在不同的计算机上运行，就要求程序设计语言能够跨越各种软件和硬件平台，而 Java 则满足了这一需求。

1995 年，美国 Sun Microsystems 公司正式向 IT 业界推出了 Java 语言，该语言具有安全、跨平台、面向对象、简单、适用于网络等显著特点。当时以 Web 为主要形式的互联网正在迅猛发展，Java 语言的出现迅速引起所有程序员和软件公司的极大关注，程序员们纷纷尝试用 Java 语言编写网络应用程序，并利用网络把程序发布到世界各地运行，包括 IBM、Oracle、微软、Netscape、Apple、SGI 等大公司纷纷与 Sun Microsystems 公司签订合同，授权使用 Java 平台技术。微软公司总裁比尔·盖茨在经过研究后认为“Java 语言是长时间以来最卓越的程序设计语言”。目前，Java 语言已经成为最流行的网络编程语言，截止到 2001 年，全世界大约有 310 万 Java 程序员，许多大学纷纷开设 Java 课程，Java 正逐步成为世界上程序员最多的编程语言。

在经历了以大型机为代表的集中计算模式和以 PC 机为代表的分散计算模式之后，互联网的出现使得计算模式进入了网络计算时代。网络计算模式的一个特点是计算机是异构的，即计算机的类型和操作系统是不相同的，如 SUN 工作站的硬件是 SPARC 体系，软件是 UNIX 中的 Solaris 操作系统，而 PC 机的硬件是 INTEL 体系，操作系统是 Windows 或者是 Linux，因此相应的编程语言基本上只是适用于单机系统，如 COBOL、FORTRAN、C、C++ 等。网络计算模式的另一个特点是代码可以通过网络在各种计算机上进行迁移，这就迫切需要一种跨平台的编程语言，使得用它编写的程序能够在网络中的各种计算机上正常运行，Java 就是在这种需求下应运而生的。正是因为 Java 语言符合互联网时代的发展要求，所以才获得了巨大的成功。



1.1 项目概述

在本项目中主要学习 Java 的基本特性、Java 的发展简史和运行机制，以及 Java 开发环境的安装、配置方法，同时通过介绍 Java 应用程序的一个简单示例来学习 Java 应用程序的开发流程。



1.2 项目目的

- 了解 Java 的基本概念、特点。
- 掌握 Java 运行环境设置和开发工具的使用。
- 掌握 Java 应用程序的编写、调试、运行。



1.3 项目支撑知识

1.3.1 项目开发背景知识 1——Java 的发展历史

1991 年, Sun 公司的 James Gosling、Bill Joe 等人为电视、烤面包机等家用电器的交互操作开发了一个 Oak(一种橡树的名字)软件,它是 Java 的前身。当时, Oak 并没有引起人们的关注,直到 1994 年, Internet 的迅猛发展与 WWW(万维网)应用的快速增长给 Java 语言带来了无限生机,用 Java 编写的浏览器 HotJava 以及 Applet 在 Web 上的应用使得 Java 逐渐成为 Internet 上受欢迎的开发与编程语言。由于涉及促销和法律的因素, 1995 年 Oak 更名为 Java。Java 的得名还有段小插曲:一天, Java 小组成员正在喝咖啡,讨论给新语言起个什么名字的问题,有人提议用 Java(Java 是印度尼西亚盛产咖啡的一个岛屿),这个提议得到了其他成员的赞同,于是他们就采用 Java 来命名此新语言。很快 Java 被工业界所认可,一些著名的计算机公司纷纷购买了 Java 语言的使用权,如 Microsoft、IBM、HP、NetScape、Novell、Apple、DEC、SGI 等,当时许多公司希望能将 Java 嵌入他们的各种开发操作系统中,于是 Sun 采用颁发许可证的办法允许各家公司把 Java 虚拟机和 Java 的 Applets 类库嵌入他们开发的操作系统中,这使得开发人员能很简便地选择多种平台使用 Java 语言编程,不同的用户也就可以脱离 Web 浏览器来运行 Java 应用程序。Java 语言被美国著名的杂志“PC Magazine”评为“1995 年十大优秀科技产品”,从此开始了 Java 应用的新篇章。

由此可见,Java 的发展速度是惊人的,下面回顾一下它的主要发展历程:

1995 年,Java 语言诞生。

1996 年,Java 的第一个开发包 JDK1.0 发布,10 个最主要的操作系统供应商申明将在其产品中支持 Java 技术。

1997 年,JDK1.1 发布。

1998 年,JDK1.1 下载量超过 200 万次;Sun 发布 JDK 1.2(也称 Java 2),Java 2 平台的发布是 Java 发展史的新里程碑。JFC/Swing 技术发布后,JFC/Swing 被下载了 50 多万次。

1999 年, Sun 公司按照 Java 技术架构,将 Java 2 平台分为 3 个版本:

(1) J2ME——Java 2 Micro Edition,适用于小型设备和智能卡的版本。

(2) J2SE——Java 2 Standard Edition,用于工作站、PC 机的标准版本。

(3) J2EE——Java 2 Enterprise Edition,适用于企业环境的版本。

2000 年,JDK1.3 发布,同年 JDK1.4 发布。

2001 年,Nokia 公司宣布到 2003 年将出售 1 亿部支持 Java 的手机,同年 J2EE1.3 发布。

2002 年,J2SE1.4 发布,至此 Java 的计算能力得到了大幅度提升,J2EE SDK 的下载量达到 200 万次。

2003 年,5.5 亿台桌面机上运行 Java 程序,75% 的开发人员将 Java 作为首要开发工具。

2004 年,J2SE1.5 发布,这是 Java 语言在其发展史上的又一里程碑事件,为了显示这个版本的重要性,J2SE1.5 更名为 J2SE 5.0。

2005 年,JavaOne 大会召开,Sun 公司公开 Java SE6,此时 Java 的各种版本被更名,取消其中的数字“2”;J2EE 更名为 Java EE,J2SE 更名为 Java SE,J2ME 更名为 Java ME。

1.3.2 项目开发背景知识2——Java 语言特性

Java 自诞生之日起就受到了全世界的关注。这是由其自身的众多突出优点造成的,具体来说有以下主要特点。

1. Java 语言简单易学

Java 语言的语法与 C 语言和 C++ 语言很接近,便于大多数程序员学习和使用。另外,Java 丢弃了 C++ 中很少使用的、很难理解的、令人迷惑的那些特性,如操作符重载、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废料收集,使得程序员不必为内存管理而担忧。

2. Java 语言是面向对象的

Java 语言提供类、接口和继承等原语,为了简单起见,它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(关键字为 implements)。Java 语言全面支持动态绑定,而 C++ 语言只对虚函数使用动态绑定。面向对象可以说是 Java 最重要的特性,它不支持类似 C 语言那样的面向过程的程序设计技术。

3. Java 语言是分布式的

Java 语言支持互联网应用的开发,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. Java 语言是健壮的

Java 的强类型机制、异常处理、废料的自动收集等是 Java 程序健壮性的重要保证。对指针的丢弃是 Java 的明智选择,Java 的安全检查机制使得 Java 更具健壮性。

5. Java 语言是安全的

Java 通常被用于网络环境中,为此 Java 提供了一个安全机制以防恶意代码的攻击。除了 Java 语言具有的许多安全特性以外,Java 对通过网络下载类具有一个安全防范机制(类 ClassLoader),如分配不同的名字空间以代替本地同名类、字节码检查,并提供安全管理机制(类 SecurityManager)让 Java 应用设置安全哨兵。在网络、分布环境下的 Java 必须防止病毒的入侵,而 Java 程序的三级代码安全检查机制即可防止非法代码的入侵,阻止对内存的越权访问,避免病毒的侵害。Java 程序在执行前,要经过三次检查,分别是 Java 本身的代码检查、字节码的检查、Java 解释器的检查,如图 1-1 所示。

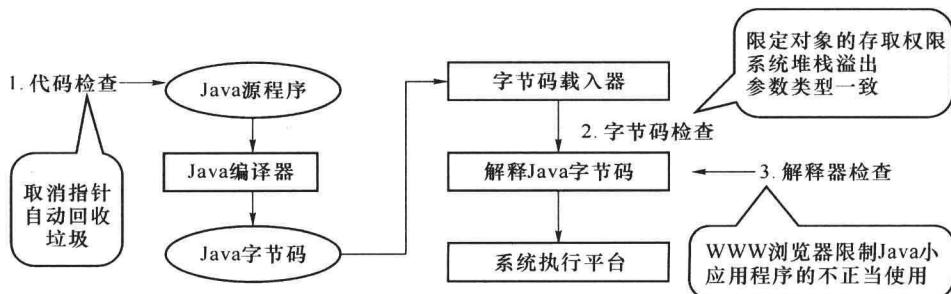


图 1-1 Java 的安全措施

6. Java 语言是体系结构中立的

Java 程序(后缀为 java 的文件)在 Java 平台上被编译为体系结构中立的字节码格式(后缀为 class 的文件),然后可以在实现这个 Java 平台的任何系统中运行,这种途径适合异构的网络环境和软件的开发。

7. Java 语言是可移植的

Java 语言的可移植性来源于体系结构的中立性。另外,Java 还严格规定了各个基本数据类型长度。Java 系统本身也具有很强的可移植性,Java 编译器是通过 Java 实现的,Java 的运行环境是通过 ANSIC 实现的。

8. Java 语言是解释型的

如上所述,Java 程序在 Java 平台上被编译为字节码格式,然后可以在实现这个 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联结阶段被载入到运行环境中。

9. Java 是高性能的

与那些解释型的高级脚本语言相比,Java 的确是高性能的。事实上,Java 的运行速度随着 JIT (Just-in-Time) 编译器技术的发展越来越接近于 C++。

10. Java 语言是多线程的

在 Java 语言中,线程是一种特殊的对象,它必须由 Thread 类或其子(孙)类来创建。通常有两种方法来创建线程:其一,使用型构为 Thread(Runnable) 的构造子将一个实现了 Runnable 接口的对象包装成一个线程;其二,从 Thread 类派生出子类并重写 run 方法,使用该子类创建的对象为线程。值得注意的是 Thread 类已经实现了 Runnable 接口,因此任何一个线程均有它的 run 方法,而 run 方法中包含了线程所要运行的代码。线程的活动由一组方法来控制。Java 语言支持多个线程的同时执行,并提供多线程之间的同步机制(关键字为 synchronized)。

11. Java 语言是动态的

Java 语言的设计目标之一是适应于动态变化的环境。Java 程序需要的类能够动态地载入运行环境,也可通过网络载入所需要的类,这有利于软件的升级。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。

Java 语言的优良特性使得 Java 应用具有无比的健壮性和可靠性,也减少了应用系统的维护费用。Java 对对象技术的全面支持和 Java 平台内嵌的 API 能缩短应用系统的开发时间并降低成本。Java 编译一次,到处可运行的特性使得其能够提供一个随处可用的开放结构和在多平台之间传递信息的低成本方式,特别是 Java 企业应用编程接口(Java Enterprise APIs)为企业计算及电子商务应用系统提供了有关技术和丰富的类库。

1.3.3 项目开发背景知识 3——Java 运行机制

Java 程序的运行必须经过编写、编译、运行三个步骤。

(1) 编写是指在 Java 开发环境中进行程序代码的输入,最终形成后缀名为 .java 的 Java 源文件。

(2) 编译是指使用 Java 编译器对源文件进行错误排查的过程,编译后将生成后缀名为 .class 的字节码文件,这不同于 C 语言(C 语言最终生成可执行文件)。

(3) 运行是指使用 Java 解释器将字节码文件翻译成机器代码,执行并显示结果,这一过程如图 1-2 所示。

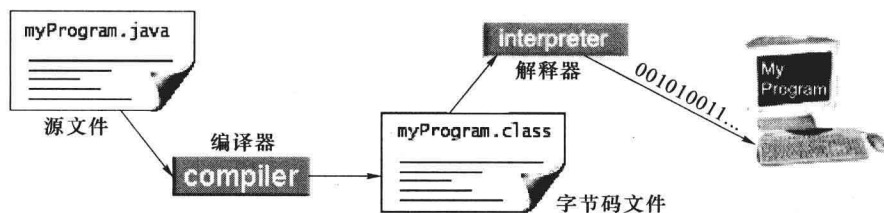


图 1-2 Java 程序运行流程

字节码文件是一种和任何具体机器环境及操作系统环境无关的中间代码,是一种二进制文件,是 Java 源文件由 Java 编译器编译后生成的目标代码文件。编程人员和计算机都无法直接读懂字节码文件,它必须由专用的 Java 解释器解释执行,因此 Java 是一种在编译基础上进行解释运行的语言。

Java 解释器负责将字节码文件翻译成具体硬件环境和操作系统平台下的机器代码,以便执行。因此 Java 程序不能直接运行于现有的操作系统平台上,它必须运行在被称为 Java 虚拟机的软件平台之上。

Java 虚拟机(JVM)是运行 Java 程序的软件环境,Java 解释器就是 Java 虚拟机的一部分。在运行 Java 程序时,首先启动 JVM,然后由其负责解释执行 Java 的字节码,并且 Java 字节码只能运行于 JVM 之上。这样利用 JVM 就可以把 Java 字节码程序和具体的硬件平台以及操作系统环境分隔开来,只要在不同的计算机上安装了针对于特定具体平台的 JVM,Java 程序就可以运行,而不需考虑当前具体的硬件平台及操作系统环境,也不需考虑字节码文件是在何种平台上生成的。JVM 把这种不同软硬件平台的具体差别隐藏起来,从而实现了真正的二进制代码级的跨平台移植。JVM 是 Java 平台无关性的基础,Java 的跨平台特性正是通过在 JVM 中运行 Java 程序实现的。Java 虚拟机的运行机制图可以通过图 1-3 予以说明。

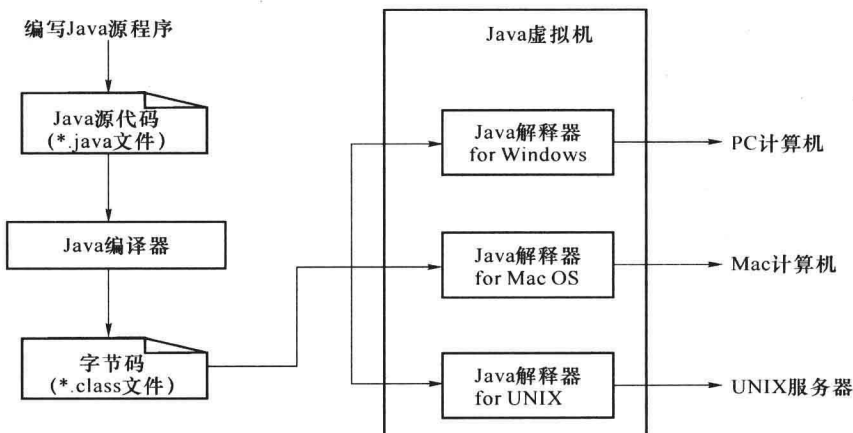


图 1-3 Java 虚拟机的运行机制

Java 语言这种“一次编写,到处运行”(write once,run anywhere)的方式,有效地解决了目前大多数高级程序设计语言需要针对不同系统来编译产生不同机器代码的问题,即硬件环境和操作平台的异构问题,大大降低了程序开发、维护和管理成本。

需要注意的是,Java 程序通过 JVM 可以达到跨平台特性,但 JVM 是不跨平台的。也就是说,不同操作系统之上的 JVM 是不同的,Windows 平台之上的 JVM 不能用在 Linux 上面,反之亦然。

1.3.4 项目开发背景知识 4——Java 开发环境

Java 开发环境的基本要求非常低,只需一个 Java 开发工具包(Java Development Kit,JDK),再加上一个纯文本编辑器即可。为了提高开发效率,可以使用功能强大的文本编辑工具,如记事本、UltraEdit 等。对于熟练的开发人员,为了进一步提高开发效率,还可以使用具有可视化功能的 Java 专用开发工具,如 Jcreator、Jbuilder、J++、NetBeans。本书程序以记事本作为编辑工具。

到目前为止,Sun 公司先后发布了多个 JDK 版本,其主要版本的特点如下:

(1) JDK1.0:于 1995 年推出,是 Java 最早的编译环境。

(2) JDK1.2:于 1998 年发布,在图形化用户界面、数据库互联以及其他许多方面作了重大改进。这个版本也被称作带有 JDK 1.2 的 Java 1.2 版本。由于该版本具有重大改变,所以从 Java 1.2 版本后的 Java 语言也被称作 Java 2 语言。

(3) JDK1.3:在多媒体应用、编译速度等方面作了改进。

(4) JDK1.4:断言(assert),为程序的调试提供了强有力的支持。

(5) JDK1.5(JDK5.0):第一个对 Java 语言作出最大改进的版本,添加了泛型类型、“for each”循环、自动打包、元数据等新特性。

(6) JDK1.6:为目前最新的版本。JDK1.6 中有九大新特性,其中之一是 JDK1.6 中自带 JAXB 2.0。JAXB 是 Java Architecture for XML Binding 的缩写,它可以将一个 Java 对象转变成为 XML 格式,反之亦然。比起 JDK1.0(JSR 31),JAXB2(JSR 222)用 JDK5 的新特性 Annotation 来标示要做绑定的类和属性等,这就极大简化了开发的工作量。由于篇幅限制,其他特性不再逐一介绍。

通常所说的 J2SDK 是指与 Java 2 语言相对应的 Java SDK(Java Software Development Kits),它是对较新版本中 JDK 的一个特定称呼。JDK 包括 Java 的编译器、解释器、调试器等开发工具以及 Java API 类库。编程人员和最终用户可以利用这些工具来开发 Java 程序。不同的操作系统(如 Windows、UNIX/Linux、Mac OS)有相应的 Java 开发包安装程序,本书使用 Windows 操作系统环境下的 Java 开发包。本书中所给的例子程序均可在 JDK 1.6.0 环境下调试通过。

1. JDK 的内容

在 JDK 根目录下,有 bin、jre、lib、demo、include 子目录和一些文件,其功能如下:

1) 开发工具

开发工具位于 bin 子目录中,是指工具和实用程序,可进行 Java 程序开发、编译、运行、调试等(有关详细信息,请参见工具文档)。

2) 运行时环境

运行时环境位于 jre 子目录中,是 Java SE 运行时环境的实现。该运行时环境包含 Java 虚拟机、类库以及其他文件,可支持执行以 Java 编程语言编写的程序。

3) 附加库

附加库位于 lib 子目录中,是开发工具需要的附加类库和支持文件。

4) 演示 Applet 和应用程序

demo 子目录中包含演示 Java Applet 和 Java 应用程序的示例,有含源代码的 Java 平台编程示例,包括使用 Swing 和其他 Java 基类以及 Java 平台调试器体系结构的示例。

5) c 头文件

c 头文件位于 include 子目录中,包含支持使用 Java 本机界面、JVM™ 工具界面以及 Java 平台的其他功能进行本机代码编程的头文件。

6) 源代码

源代码位于 src.zip 中,包含组成 Java 2 核心 API 的所有类的 Java 编程语言源文件(即 java.*、javax.* 和某些 org.* 包的源文件,但不包括 com.sun.* 包的源文件)。此源代码仅用于提供信息,以便帮助开发者学习和使用 Java 编程语言。

2. JDK 的基本命令

JDK 包含用于开发和测试以 Java 编程语言编写并在 Java 平台上运行程序的工具。这些工具被设计为从命令行使用。除了 appletviewer 以外,这些工具不提供图形用户界面。JDK 的基本命令包括 javac、java、jdb、javap、javadoc、appletviewer。

(1) javac:Java 编译器,用来把 Java 程序编译成字节码。

命令格式:javac [选项] 源程序名

(2) java:Java 解释器,执行已经转换成字节码的 Java 应用程序。

命令格式:java [选项] 类名 [参数]

(3) jdb:Java 调试器,用来调试 Java 程序。

启动 jdb 的方法有两种:第一种方法格式与 Java 解释器类似;第二种是把 jdb 附加到一个已运行的 Java 解释器上,该解释器必须是带 -debug 项启动的。

(4) javap:反编译,将类文件还原回方法和变量。

命令格式:javap [选项] 类名

(5) javadoc:文档生成器,创建 HTML 文件。

命令格式:javadoc [选项] 源文件名

(6) appletviewer:小应用程序 Applet 浏览工具,用于测试并运行 Applet。

命令格式:appletviewer [选项] URL

其中,URL 是包含被显示 Applet 的 HTML 文件的统一资源定位符,当 HTML 文件位于本地机上时,只需写出文件名。



1.4 项目实施

1.4.1 Java 开发环境的搭建

一、JDK 安装

为了建立 Java 的运行环境,可以到官方网站下载最新的 JDK。建议同时下载其 Java Documentation,这是 Java 帮助文档。下面以 JDK1.6.0 版的 J2SDK 为例(即 j2sdk 1.6.0),运行下载的 j2sdk-6u19-windows-i586.exe 文件,进入 JDK 的安装界面。安装程序会弹出许可证协议阅读画面,如图 1-4 所示,单击【接受A】按钮后,进入 JDK 的默认安装界面,如图 1-5 所示。

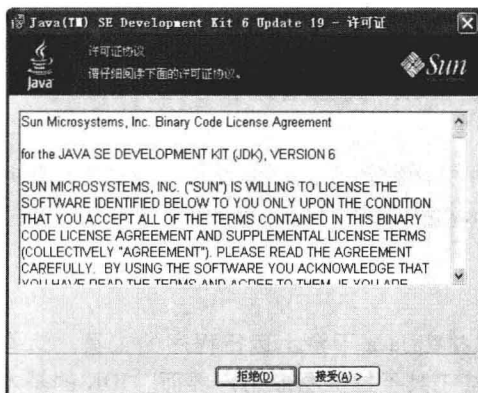


图 1-4 许可证协议阅读界面

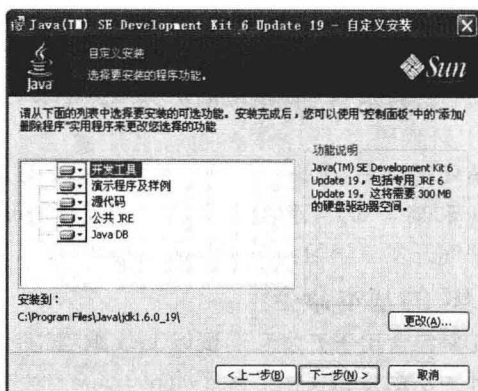


图 1-5 JDK 默认的安装界面

默认安装于 C:\Program Files\Java\jdk1.6.0 目录下,用户可以更改这个默认安装目录,单击【更改(A)⋯】按钮,进入如图 1-6 所示的更改 JDK 安装路径的界面,将路径确定为 D:\Java\jdk1.6.0,单击【确定】按钮后回到如图 1-7 所示的安装界面。

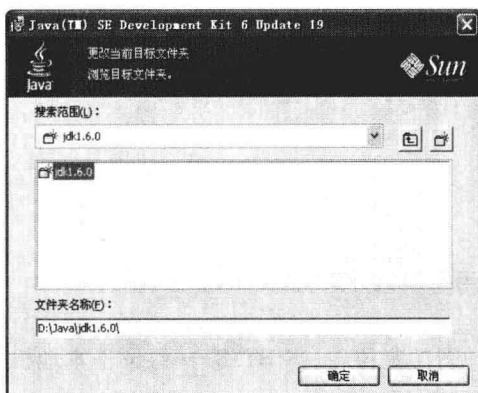


图 1-6 更改 JDK 安装路径

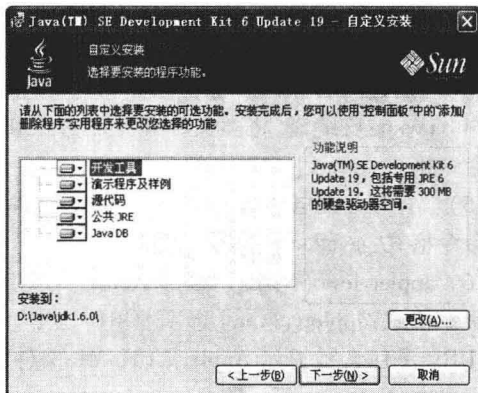


图 1-7 更改安装路径后的 JDK 安装界面

单击【下一步(N)】按钮后,JDK 安装程序运行界面如图 1-8 所示。在 JDK 安装程序的运行中,会弹出 JRE 的安装界面,如图 1-9 所示,初学者可以单击【取消】按钮,不选择安装。

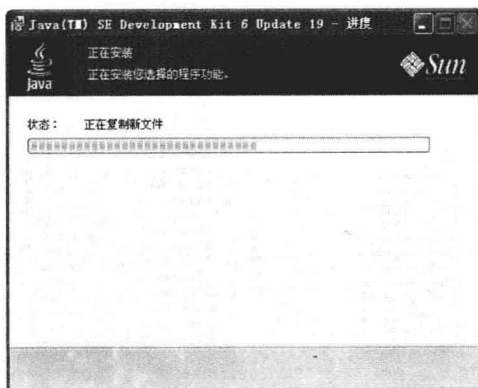


图 1-8 安装程序运行界面

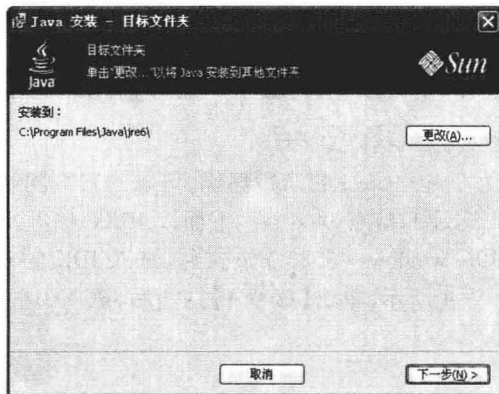


图 1-9 安装程序转入 JRE 安装界面