

Java语言 与面向对象程序设计

● 朱福喜 尹为民 余振坤 编著

<http://www.com>

<http://www.com>



全国优秀出版社
武汉大学出版社

Java 语言与面向对象程序设计

朱福喜 尹为民 余振坤 编著

武汉大学出版社

内 容 简 介

Java 语言目前不仅是一门最为流行的计算机语言,而且形成一种专门的技术。本书从 Java 的基本概念入手,介绍了 Java 语言的基础知识与主要功能,系统分析了 Java 语言的编程机制,并用软件工程的理论和方法,阐述了使用 Java 语言的编程技巧。本书还介绍了一些 Java 语言的高级特性,如网络编程、数据库连接、多线程以及远程对象调用等。可广泛适用于计算机及相关专业的本科生和研究生作为学习 Java 语言的教材,也适合软件开发人员及其他人员参考学习。

图书在版编目(CIP)数据

Java 语言与面向对象程序设计/朱福喜,尹为民,余振坤编著. —武汉:
武汉大学出版社,2002.12

ISBN 7-307-03784-X

I. J… II. ①朱… ②尹… ③余… III. ①Java 语言—程序设计
②面向对象语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 106379 号

责任编辑:郭志安 责任校对:黄添生 版式设计:支 笛

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:wdp4@whu.edu.cn 网址:www.wdp.whu.edu.cn)

印刷:安陆市鼎鑫印务有限责任公司

开本:787×1092 1/16 印张:20.5 字数:496 千字

版次:2002 年 12 月第 1 版 2002 年 12 月第 1 次印刷

ISBN 7-307-03784-X/TP·132 定价:28.60 元

版权所有,不得翻印;凡购我社的图书,如有缺页、倒页、脱页等质量问题,请与当地图书销售部门联系调换。

前 言

Java 语言自 1995 年诞生以来,在短短的几年时间内以迅猛的速度席卷全球,它不仅是一门最为流行的计算机语言,而且形成一种专门的技术,带来了 20 世纪末和 21 世纪初网络科技和网络经济超常发展。Java 与平台的无关性、高可靠性、安全机制和内嵌的网络支持使她成为编写网络应用程序的首选工具之一,并被誉为“网络上的世界语”。Java 正像她的象征符号“一杯冒着热气的香浓咖啡”那样,既让人爱不释手,又令人回味无穷。

面向对象技术被称为是程序设计方法学的一场革命,它已逐步取代传统的面向过程的程序设计技术,成为应用软件开发领域的主流技术。它所倡导的面向现实世界的思维方式、自然简洁的设计风格符合现代大规模软件开发的要求和潮流。这种技术已成为软件开发人员的最基本的素养之一。而 Java 语言属于纯面向对象语言,是面向对象技术成功应用的一个著名范例。本书以 Java 为范型,介绍了面向对象程序设计的主要原理和方法,同时也介绍了 Java 的最主要的核心技术,并作了较深入的讨论。

本书从 Java 的基本概念入手,借助于 Java 语言来讲授面向对象程序设计,进行面向对象的思维方法的训练,通过学习、设计及实现,使读者能掌握其方法、原则与理论,具有一定的面向对象设计、开发能力,为大型软件的研究、设计打下基础。在此基础上,同时也本书还系统分析了 Java 语言的编程机制,阐述 Java 语言的精髓和使用 Java 语言的编程技巧,同时也介绍了一些 Java 语言的高级特性,如多线程、网络编程、数据库连接以及远程对象调用等。书中列举了大量的程序实例,并对内容和例题选择都作了严格的控制,既保证突出阐明原理和方法,又保证有一定的实用性,同时也确保了一定的广度和深度,在难易程度上遵循由浅入深、循序渐进的原则。在举例过程中不仅注意到让读者能从实例运用中举一反三,同时也注意其规模适中,不大不小。这些程序实例都尽可能地站在读者的角度,体会 Java 语言的精髓以及直接根据这些源程序来快速编写 Java 程序,直接切入有关应用。

本书第一章主要介绍了 Java 的发展、语言特点和展示一个很小的实例,使读者对 Java 有一个概貌性的了解。第二章介绍了面向对象编程基础,即面向对象的一般原理和方法。第三章介绍了 Java 编程的基础知识,主要包括数据类型、变量、表达式和流程控制语句。第四、五两章介绍了 Java 面向对象编程的知识,以类为中心详细地讨论了面向对象技术的四大特点:封装、抽象、继承和多态及其在面向对象程序设计中的具体应用。无论是从 Java 语言的角度还是从面向对象技术的角度,这两章都是核心,通过这两章的学习,读者可以着实掌握 Java 语言和面向对象编程的精髓。第六章介绍流和文件,这一章不仅是文件和输入输出操作的基础,也是后续的 Java 高级编程,如网络编程的基础。第七章介绍 Java 图形用户界面的设计和编程实现,通过这章的学习,读者可以编写出丰富多彩的程序界面,使开发出的应用程序有一个很好的“第一印象”。第八章介绍 Java 的多线程编程和异常处理,掌握这一章的内容可以编写出功能复杂的程序又能够保证程序有很好的强壮性。第九章介绍 Ap-

plet 的设计,Applet 能够使 Java 语言在全球网上充分展示自己的魅力。第十章介绍网络编程,这一章能够充分显示 Java 的强大网络编程功能。第十一章介绍 Java 数据库连接(JDBC),这一章可以很方便地在 Java 程序中引入数据库应用。第十二章介绍 Java 远程方法调用(RMI),有了这一章的知识,我们可以方便地使用从事分布计算,这也是 Java 非常引人注目的特色之一。

本书可广泛适用于计算机及相关专业的本科生和研究生作为学习 Java 语言的教材,也适合其他有关软件开发人员作为自学的参考书,同时也可用作培训教程。

本书第一章至第七章由朱福喜和尹为民共同编写,第八章至第十章由朱福喜编写,第十一章和第十二章由余振坤和朱福喜共同编写,全书最后由朱福喜统稿,由余振坤验证本书的所有实例。

在本书的编写过程中,作者引用了陈劲欣的毕业论文中的实例。在编排录入工作中得到了王瑛、朱丽达、韩君的大力支持。同时,本书编写得到了武汉大学出版社电子音像出版部的大力支持。在此谨向他们表示衷心感谢。

由于时间和水平所限,难免有错,恳请读者批评指正,使本书得以改进和完善。

作 者

2000 年 8 月于珞珈山

目 录

第一章 Java 概述	1
1.1 Java 技术的出现与形成	1
1.2 语言的特色	2
1.3 Java 的开发和执行环境	4
1.3.1 JDK 的下载	4
1.3.2 JDK 的安装与环境配置	4
1.3.3 JDK 环境工具	5
1.4 一个简单的 Java 程序	7
1.5 Java 独立应用程序的基本结构剖析	9
第二章 面向对象的基础知识	12
2.1 面向对象问题求解概述	12
2.2 对象	13
2.3 消息	14
2.4 类	15
2.5 实例成员与类成员	16
2.6 继承	16
2.6.1 继承的定义	16
2.6.2 继承的优越性	17
2.7 抽象与封装	18
2.7.1 抽象	18
2.7.2 封装	18
2.8 多态性	19
第三章 Java 语言基础	21
3.1 一个简单的 Java 程序举例	21
3.2 变量和数据类型	22
3.2.1 变量名字	22
3.2.2 变量类型	22
3.2.3 变量作用域	24
3.2.4 变量初始化	25
3.2.5 final 变量	25

3.3 运算符	26
3.3.1 算术运算符	26
3.3.2 关系与逻辑运算符	30
3.3.3 位运算符	32
3.3.4 赋值运算符	33
3.3.5 其他运算符	34
3.4 表达式与语句	35
3.4.1 表达式	35
3.4.2 语句	36
3.5 控制语句	36
3.5.1 if 语句	37
3.5.2 switch 语句	38
3.5.3 while 和 do-while 语句	39
3.5.4 for 语句	41
3.5.5 跳转语句	42
3.6 数组	45
3.6.1 创建和使用数组	46
3.6.2 对象数组	47
3.6.3 多维数组	48
3.6.4 复制数组	49
第四章 类与对象	51
4.1 类的建立与使用	51
4.1.1 类的定义	51
4.1.2 创建对象与定义构造函数	53
4.1.3 类声明与类体的描述	55
4.1.4 成员变量的声明	57
4.1.5 方法的声明与实现	58
4.2 修饰符	66
4.2.1 类的修饰符	66
4.2.2 域修饰符	67
4.2.3 方法修饰符	67
4.2.4 访问控制符	70
4.3 包	75
4.3.1 包的创建	76
4.3.2 包的使用	77
4.4 类的其他说明	78
4.4.1 实例成员和类成员	78
4.4.2 嵌套类	81

4.4.3 撤销方法与清理无用对象	82
第五章 继承与多态	84
5.1 子类、父类与继承机制	84
5.1.1 继承的概念	84
5.1.2 类的层次	85
5.2 Java 的继承	86
5.2.1 子类的定义	86
5.2.2 域的继承与隐藏	88
5.2.3 方法的继承与覆盖	90
5.3 多态性	92
5.3.1 多态性的概念	92
5.3.2 覆盖实现多态性	93
5.3.3 重载实现多态性	94
5.4 构造函数的继承与重载	96
5.4.1 构造函数的重载	96
5.4.2 构造函数的继承	97
5.5 接口	101
5.5.1 接口与多继承	101
5.5.2 接口的定义	102
5.5.3 接口的实现	103
5.5.4 接口的使用	104
第六章 流和文件	106
6.1 I/O 流概述	106
6.1.1 简单的 I/O 流	108
6.1.2 过滤流	108
6.1.3 其他 I/O 类和接口	108
6.2 基本 I/O 流的使用方法	109
6.2.1 InputStream 类	109
6.2.2 OutputStream 类	111
6.2.3 PrintStream 类	111
6.2.4 管道流	112
6.2.5 SequenceInputStream 类	113
6.3 文件处理与随机访问	113
6.3.1 文件与目录管理	113
6.3.2 文件输入输出流	115
6.3.3 随机流访问	117
6.4 过滤流的使用	117

6.4.1	BufferedInputStream 类和 BufferedOutputStream 类	119
6.4.2	DataInputStream 类和 DataOutputStream 类	119
6.4.3	LineNumberInputStream 类	121
6.4.4	PushbackInputStream 类	121
6.5	对象流	121
6.5.1	保存对象数据	121
6.5.2	检索对象数据	123
第七章	Java 的用户界面	126
7.1	图形用户界面概述	126
7.2	基本的图形用户界面程序	127
7.2.1	几个基本的类	127
7.2.2	事件响应	129
7.2.3	简单的 GUI 输入输出	132
7.3	事件处理	134
7.3.1	基本事件	135
7.3.2	Java 中的事件处理	135
7.4	GUI 构件和布局管理	141
7.4.1	布局管理器、面板和标签	141
7.4.2	List 和 TextArea	147
7.5	菜单和对话框	152
7.5.1	Menu、MenuBar 和 MenuItem 类	152
7.5.2	对话框	154
7.6	图形与图形的绘制	156
7.6.1	Graphics 类	157
7.6.2	简单绘图	158
7.6.3	Canvas	161
7.6.4	字体(Font)类	166
第八章	多线程与异常处理	169
8.1	多线程的基本概念	169
8.1.1	多任务	169
8.1.2	线程与多线程	169
8.1.3	Java 对多线程的支持	170
8.1.4	线程的状态	170
8.1.5	线程的优先级	172
8.2	线程的使用方法	172
8.2.1	通过继承 Thread 类构造线程	172
8.2.2	通过 Runnable 接口的实现构造线程	179

8.3 线程的同步	181
8.3.1 使用多线程造成的数据崩溃	181
8.3.2 同步线程	187
8.4 异常处理	193
8.4.1 Java 中出错类型	193
8.4.2 异常的抛出	196
8.4.3 异常的处理	197
8.4.4 finally 子句	199
第九章 Applet	201
9.1 Applet 基础	201
9.1.1 Applet 类的定义	201
9.1.2 Applet 类的实例变量	202
9.1.3 Applet 类的成员方法	202
9.1.4 Applet 的生命周期	203
9.1.5 从独立应用程序到一个 Applet 的转换	205
9.1.6 确保 Applet 正常运行	206
9.2 Java 的 <APPLET> 标记	207
9.2.1 <APPLET> 标记属性	207
9.2.2 利用标记参数向 Applet 传递信息	210
9.3 Applet 应用	215
9.3.1 Applet 与图形用户接口(GUI)	215
9.3.2 Applet 与线程	218
9.3.3 Applet 编程技巧	218
9.4 利用 AppletViewer 调试 Applet	223
第十章 网络程序设计	227
10.1 基本概念	227
10.1.1 IP 地址	227
10.1.2 端口	228
10.1.3 套接字	228
10.1.4 Internet 协议	229
10.2 使用 URL 访问 WWW 资源	231
10.2.1 创建一个 URL	231
10.2.2 直接从 URL 读取内容	232
10.2.3 建立一个 URL 连接并从中读取内容	233
10.3 在网络上实现客户和服务端	235
10.3.1 创建客户程序	235
10.3.2 编写服务器应用程序	245

10.4 基于客户/服务器模式的分布计算	252
10.4.1 分布模型	252
10.4.2 并行应用的实现框架	253
10.4.3 并行计算实例	254
第十一章 Java 数据库连接	263
11.1 JDBC 概述	263
11.1.1 JDBC 简介	263
11.1.2 JDBC 与 ODBC	263
11.1.3 JDBC 的实现及其驱动程序	264
11.2 JDBC 预备知识	265
11.2.1 SQL 的基本内容	265
11.2.2 存储过程	268
11.3 JDBC API 介绍	269
11.3.1 连接管理	270
11.3.2 数据库访问	270
11.3.3 数据类型和数据库元数据	271
11.3.4 异常和警告	271
11.4 JDBC API 的使用方法	271
11.4.1 JDBC API 工具	271
11.4.2 连接数据库	273
11.4.3 使用 JDBC 创建基本表和视图	280
11.4.4 使用 JDBC 执行数据库查询	282
11.4.5 使用 JDBC 更新数据库	286
第十二章 RMI-Java 远程方法调用	291
12.1 概述	291
12.2 RMI 的基础知识	292
12.2.1 实现一个 RMI 应用实例	292
12.2.2 实现一个 RMI 应用的基本步骤及其分析	296
12.3 RMI 体系结构与实现机制	302
12.3.1 RMI 体系结构	302
12.3.2 RMI 具体实现机制	304
12.4 RMI 的高级功能	305
12.4.1 RMI 的安全性	306
12.4.2 传递对象和动态类加载	307
12.4.3 远程对象激活	308
12.4.4 分布式垃圾收集	312
12.4.5 RMI 的发展新方向	312
附录 JDK 环境工具及其参数	314

第一章 Java 概述

Java 是 Sun Microsystem 公司研制的一种新型的程序设计语言。在高级语言已经发展到非常丰富的今天,Java 语言能够脱颖而出,成为当今最流行的语言之一,有其历史背景和独树一帜的品质。

1.1 Java 技术的出现与形成

Java 语言的起源可追溯到十多年前,一些敢于创新的软件工程师试图开发一种可移植软件来控制家用电器,这类电器诸如:烤箱、电视、冰箱、录像机、灯光设备、电话、呼机、传真机等。1994 年 4 月,由美国 Sun Microsystem 的 Patrick Nawghton, Jame Gosling 以及 Mike Sheridan 等人组成的开发小组,开始了代号为“Green”的项目研制。Green 项目的目标是研制一种开发家用电器的逻辑控制系统,实现对家用电器的集成控制。这个产品的名称为“Oak”。但是,这个在技术上非常成功的产品,当时并未获得在商业上的成功。

1994 年末,由于 Internet 的迅猛发展,WWW 以极快的速度风靡全球。Green 项目小组发现他们的新型编程语言 Oak 比较适合于 Internet 程序的编写,于是他们结合 WWW 的需要,对 Oak 进行改进和完善,并获得了极大的成功。1995 年 1 月,Oak 被更名为 Java。这个名字的产生,既不是根据语言本身的特色来命名,也不是由几个英文单词的首字母拼成,更不是由人名或典故而来,而是来自于印度尼西亚一个盛产咖啡的岛屿,中文名叫爪哇,意为世人端上一杯热咖啡。许多程序设计师从所钟爱的热腾腾的香浓咖啡中得到了灵感,因而热腾腾的香浓咖啡也就成为 Java 语言的标志。

1995 年 5 月 23 日 Java 正式公布,以后人们对 Java 的兴趣和重视证明了这项技术将是主宰网络计算的未来。

1995 年一些著名的公司都纷纷购买语言使用权,如:IBM, Microsoft, Netscape, Novell, Apple, DEC, SGI 等。

1996 年 Java 应用软件纷纷问世,赢得软件工业界的广泛支持。

1997 年 Java 技术用于网络计算,从业界的事实标准走向法律标准。

今天,Java 已经形成一门技术,主要体现在:

- Java 软件技术:Java JDK, Java Runtime Environment, Hotjava, Java OS, JDBC, Java Beans。
- Java 虚拟机及规范,JRE。
- Java 嵌入技术:Java 芯片(如 MicroJava701),基于 Java 技术的 NC, Java Station, WebTV (机顶盒), Java 汽车, Java 手机, Java 掌上机等。
- Java Computing:基于处理功能的综合处理系统,如 J2EE Sever 或 Web Application Sever,其典型产品有 Weblogic、Webspher 等。

1.2 语言的特色

在 Sun 公司的白皮书中,对 Java 的定义是:

“Java: A Simple, object-oriented, distributed robust, secure, architecture-neutral, portable, high-performance, multi-threaded and dynamic language.”

即,Java 是一种简单的、面向对象的、分布式的、强壮的、安全的、体系结构中立的、可移植的、高性能的、多线程的和动态的语言。

这个定义充分说明了语言的特点。下面我们将简述这些特点。

1. 简单性

Java 是一种简单的语言。语言的设计者尽量把语言的构造规模变小,通过提供最基本的方法来完成指定的任务,使用者只需要理解一些基本概念,就可以编出合适的应用程序。Java 取消了许多语言中十分繁琐和难以理解的内容,例如 C++ 的指针、运算符重载、类的多继承等,并且通过实现自动垃圾收集,大大减化了程序设计者的内存管理工作。Java 在外观上让大多数程序员感到很熟悉,便于学习。同时 Java 的编译器也很小,便于在各种机型上实现。

2. 面向对象

Java 是一种面向对象的语言。这里的对象是指应用程序的数据及其操作方法。Java 的程序设计集中于对象及其接口,Java 提供了简单的类机制以及动态的接口模型,实现了模块化和信息封装。Java 类提供了一类对象的原型,并且通过继承机制,实现了代码的重用。

3. 分布性

Java 是一个分布式语言。它有一个通讯及其他功能很齐全的程序库,可以处理 TCP/IP 及其他协议,用户可用 URL 地址在网络上很方便地访问其他对象。利用 Java 来开发分布式的网络程序是 Java 的一个主要应用之一。

4. 健壮性

用 Java 编写的程序能够在多种情况下稳定执行,因为它在编译和运行时都要对可能出现的问题进行检查。Java 有一个专门的指针模型,它的作用是排除内存中的数据被覆盖和毁损的可能性。Java 还通过集成面向对象的例外处理机制,在编译时提示可能出现但未处理的例外,以防止系统的崩溃。

5. 安全性

Java 是一种安全的网络编程语言,不支持指针类型,一切对内存的访问都必须通过对象的实例来实现。这样能够防止他人使用欺骗手段访问对象的私有成员,也能够避免在指针操作中易产生的错误。此外,Java 有多个阶层的互锁保护措施,能有效地防止病毒侵入和蓄意的破坏行为。

6. 体系结构中立

Java 编译器能够产生一种与计算机体系结构无关的字节指令(Byte Code),只要安装了 Java 虚拟机,Java 就可以在相应的处理机上执行。这个过程可以用图 1.1 说明。

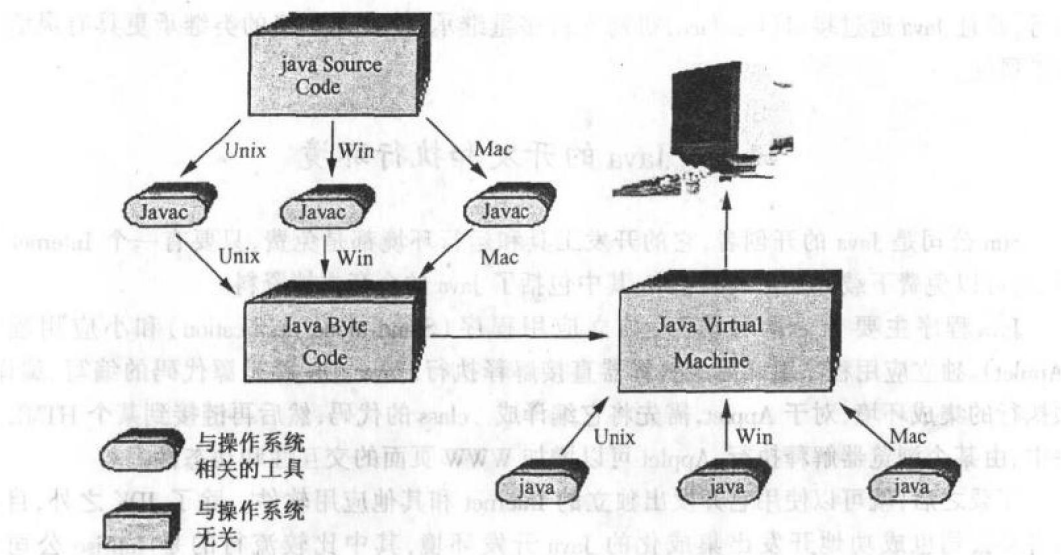


图 1.1 Java 程序与虚拟机

7. 可移植性

Java 具备有很好的移植性,这主要得益于它与平台无关的特性。同时,Java 的类库中也实现了与平台无关的接口,这使得这些类库也能移植。同时,Java 编译器主要是由 Java 本身来实现的,Java 的运行系统(解释器)由标准 C 实现,因而整个 Java 系统都具有可移植性。

8. 解释型语言

Java 解释器能直接在任何机器上解释执行 Java 字节码,字节码本身带有许多编译信息,使得连接过程更加简单。

9. 高效性

Java 的字节码能够迅速地转换成机器码,从而可以充分地利用硬件平台资源,得到较高的整体性能。

10. 多线程机制

Java 具有多线程机制,这使得应用程序能够并行地执行。它的同步机制也保证了对共享数据的共享操作,而且线程具有优先级的机制,有助于分别使用不同线程完成特定行为,

也提高了交互的实时响应能力。

11. 动态性

Java 比 C++ 语言更有动态性,更能适应不断变化的环境。Java 不会因程序库的更新而重新编译程序,所以,在类库中可以自由地加入新的方法和实例变量,不会影响用户程序的执行,并且 Java 通过接口(interface)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。

1.3 Java 的开发和执行环境

Sun 公司是 Java 的开创者,它的开发工具和运行环境都是免费,只要有一个 Internet 账号,就可以免费下载开发工具集 JDK,其中包括了 Java 的全套文档资料。

Java 程序主要分为两种类型:独立应用程序(Stand-alone application)和小应用程序(Applet)。独立应用程序编译后由解释器直接解释执行,Java JDK 就是源代码的编写、编译、预执行的集成环境,对于 Applet,需先将它编译成 .class 的代码,然后再链接到某个 HTML 文件中,由某个浏览器解释执行,Applet 可以增加 WWW 页面的交互性和动态性。

下载之后,就可以使用它开发出独立的 Internet 和其他应用软件。除了 JDK 之外,目前有许多公司也成功地开发出集成化的 Java 开发环境,其中比较流行的是 Inprise 公司的 Jbuilder 和 MicroSoft 的 Visual J++,它们都是商业版的 Java。

1.3.1 JDK 的下载

打开一个 WWW 浏览器后,可以在 Sun 公司的 Internet 站点找到 JDK 的最新版本,自由下载。如果从其他镜像站点下载 JDK,则要注意这些镜像站点是否保存的是 JDK 的最新版本。

当前的 JDK 的最高版本是 JDK1.3,它的下载网址是:

<http://Java.Sun.Com/products/JDK1.3/download-windows.html>

当然,这是支持 Windows 95/98/2000/NT 操作系统的版本。Sun 也提供支持类似 Solaris SPARC 和 Macintosh 操作系统的版本。用户可以根据操作系统平台选取合适的 JDK 版本进行下载。

下载了必要的软件之后,就可以按文档 README.TXT 介绍的安装过程,安装开发环境 JDK。

1.3.2 JDK 的安装与环境配置

下面以 Window 2000/NT 环境为背景,讲述 JDK 的安装与环境配置。

在 Window 环境下,直接点击所下载 JDK 压缩文件的图标,即可运行该文件,因为它是一个自解压的 EXE 文件。解压后,自动进入安装过程,此时可以按照提示过程,逐步完成安装。

安装时要选择一个目录,如果选择的安装目录为 C:\JDK13,则安装完毕后,用 DIR 命令显示该目录,可以显示该目录的子目录有:

BIN, LIB, DEMO, classES, INCLUDE, JRE 等。其中, BIN 目录是一些执行文件, Java 的编译程序就在该目录下。LIB 目录保存库文件。DEMO 目录是一些 APPLET 的实例。classES 目录保存的是 Java 类库。

在 C:\JDK13 目录下的文件有:

COPYRIGHT, LICENSE, README.HTML, README.TXT 等。

在 Window 2000 下, 环境变量的配置可以从“我的电脑”(右键)→高级→环境变量, 再在其中选择“系统变量”的“新建”, 然后在变量名中输入: “classPATH”, 在变量值输入: “C:\JDK13\BIN;C:\JDK13\LIB\dt.jar;C:\JDK13\LIB\TOOLS.jar”, 最后选择确定。

如果熟悉 DOS 命令, 可直接编辑自动批处理文件 Autoexec.bat 文件, 在该文件中添加设置语句:

```
set classPATH = C:\JDK13\BIN;C:\JDK13\LIB\dt.jar;C:\JDK13\LIB\TOOLS.jar;
配置完成后, 需重新启动计算机, 环境变量方能生效。
```

1.3.3 JDK 环境工具

本节将简单介绍 JDK 环境工具, 包括 Java 编译器, Java 解释器, Applet 程序观察器, Java 文档生成器, Java 调试器, C 文件生成器等。

1. 编译器——javac

javac 的作用是将源程序(.java 文件)编译成字节码(.class 文件)。Java 源程序的后缀名必须是 java。javac 一次可以编译一个或多个源程序, 对于源程序的每个类定义, 都会生成一个单独的类文件。因此, Java 源文件与生成的 class 文件之间并不存在一一对应的关系。例如, 如果在 test.java 中定义了 A, B, C 三个类, 则经过 javac 编译后要生成 A.class, B.class, C.class 三个类文件。

javac 的调用格式为:

```
javac [选项] 源文件名表
```

其中源文件名表是多个带 .java 后缀的源文件名; 选项参见附录中 javac 选项表。

2. Java 的语言解释器——Java(.exe)

java 命令解释执行 Java 字节码。格式:

```
java [选项] 类名 <参数表>
```

这里的类名代表要执行的程序, 即由生成的带 .class 后缀的类文件名, 但在上述命令中不需要带后缀。这个类名必须是一个独立程序(不能是 Applet), 其中必须带有一个按如下格式声明的 main 方法:

```
public static void main(String [ ] args) {...}
```

并且包含方法的类名必须与类文件名相同, 即与现在命令行中的“类名”相同。

在上述执行命令执行过程中, 类名后的参数表直接传递给类中的 main 方法。java 执行 main() 方法后, 若没有创建多线程, 则自动退出, 否则, 直到最后一个线程结束后才退出。若在 Java 程序中, 用户定义了新类, 则需要由 classpath 指明其存放路径。

java 命令所使用的选项参数参见附录中 java 选项表。

3. Java 语言调试工具——jdb

jdb 试用语言编写的程序。格式：

```
jdb [选项] 类名
```

或

```
jdb [-host 主机名]-password 口令
```

jdb 装载指定的类,启动自己内嵌的一个 Java 解释器,然后等待用户发出相应的调试命令,通过使用 Java debugger API 能够对本地或远程的 Java 解释器进行调试。

如果使用第一种命令格式,那么由 jdb 解释执行被调试的类。若使用第二种格式,则 jdb 将被嵌入到一个正在运行的 Java 解释器之中,这个解释器必须事先用 -debug 选项启动,而且要求用户输入一个口令,这个口令也就是出现在命令行中的口令。如果使用了 host 选项,那么 jdb 就可以嵌入到网络中由命令行中的“主机名”所指出的主机正在运行的解释器之中。

所有的调试命令参见附录中调试命令表。

4. Java 文档生成器——javadoc

javadoc 从 java 源文件生成 HTML 格式的 API 文档,内容包括类和接口的描述、类的继承层次以及类中任何非私有域的索引和介绍。格式：

```
javadoc [选项] [包名|文件名]
```

其中选项参见附录中 javadoc 选项表。

用户可以用包名或一系列的 Java 源程序名作参数。调用时,javadoc 可以自动对类、界面、方法和变量进行分析,然后为每个类生成一份 HTML 文档,并为类库中的类生成一份 HTML 索引。

5. C 头文件和源文件生成器——javah

javah 命令从一个 Java 类中生成实现 native 方法所需的 C 头和存根文件(.h 文件和 .c 文件),利用这些文件可以把 C 语言的源代码装到 Java 应用程序,使 C 可以访问 Java 对象的实例变量。格式：

```
javah [选项] 类名
```

在默认的情况下,javah 为每一个类生成一个文件,保存在当前目录中。若使用 -stub 选项则生成源文件,使用 -o 选项则把所有类的结果存于一个文件之中。

6. Java Applet 观察器——appletviewer

appletviewer 命令使用户不通过 Web 浏览器也可以观察 Applet 运行情况。格式：

```
appletviewer [-debug] HTML 文件
```

appletviewer 下载并运行 HTML 文件中包含的 Applet,如果 HTML 文档中不包含任何 Applet,appletviewer 则不采取任何行为。如果上述命令中使用了 -debug 选项,则 appletviewer 将从 jdb 内部启动,这样就可以调试 HTML 文件所引用的 Applet。