

Java 编程指南

刘德意 王峻岭 编著



清华大学出版社



Java 编程指南

刘德意 王峻岭 编著

清华大学出版社

(京)新登字 158 号

35181/35
内 容 简 介

Java 是一种由 Sun Microsystems 公司开发的新型计算机编程语言。它吸收了现有的各种语言的优点,从设计初始就与 WWW 网络紧密结合,具有平台独立性,且更安全,易学易用。本书将从 Java 的基础知识到深入的概念,由浅入深地讲解 Java 的编程方法,并附以准确详尽的实例。

本书适合大专院校师生学习参考,也可作为培训教材。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Java 编程指南/刘德意,王峻岭编著. —北京:清华大学出版社,1996.12
ISBN 7-302-02397-2

I. J… II. ①刘… ②王… III. Java 语言-程序设计-指南 IV. TP312Ja-62

中国版本图书馆 CIP 数据核字(96)第 24492 号

出 版 者:清华大学出版社 (北京清华大学校内,邮政编码:100084)

印 刷 者:北京市清华园胶印厂

发 行 者:新华书店总店科技发行所

开 本:787×1092 1/16 印张:12 字数:284 千字

版 次:1997 年 2 月第 1 版 1997 年 2 月第 1 次印刷

书 号:ISBN 7-302-02397-2/TP·1201

印 数:00001—10000

定 价:14.50 元

前 言

Java^(TM)是由美国 Sun Microsystems 公司开发的第一种能在 Internet 上具有“硬件/软件独立性”和交互能力的新型程序设计语言。Java 程序与平台无关，并且可以分布式地在远地运行。它使得开发者编写的软件可以在网上提供给用户，用户在自己的计算机上以 WWW 页面作为取回和执行软件的平台。

“Java”是美国 Sun Microsystems 公司 Java 开发小组历经无数次的会议之后才被选出的。生动性 (liveliness)、动画能力 (animation)、速度 (speed)、交互性 (interactivity) 是当初选择名字时所欲表达出的 Java 的特色。“Java”是在无数的建议中脱颖而出的，它并不是由几个单字的首字母所组成，而是从许多程序设计者所钟爱的热腾腾、香浓咖啡中产生的灵感。

Java 一经推出即引起世界各种媒体的注意与热烈地参与讨论，如 Time、BusinessWeekly、CNN 等，可见其魅力。事实上，以 Java 的功能，势必在信息服务中掀起另一番高潮，网络的发展将被带入另一个新的纪元。

随着 Internet 的发展，Java 受到越来越多的人的关注。本书将通过详尽的实例和简明的讲解，由浅入深地带您步入 Java 的世界。本书共分九章，第一章介绍 Java，第二章介绍 Java 的运行机制及其环境，第三、四、五章介绍 Java 的基本概念，第六章介绍 Java 的应用——AWT 的使用，第七、八章进一步介绍 Java 的相关知识——多线程和 I/O，第九章介绍专用于 WWW 页面的 Java 小应用程序 Applet 的编写和使用。

由于时间和人员所限，书中难免有所疏漏，望读者原谅并予以指正。我们愿意与您共同讨论有关 Java 的技术问题。

目 录

第一章 Java介绍	1
1.1 什么是Java	1
1.2 Java对Internet的影响和作用	1
1.3 Java的特点	2
1.4 Application和Applet	4
1.5 小结	4
第二章 Java的运行机制及其环境	5
2.1 Java的运行机制	5
2.2 Java的开发运行环境	6
2.2.1 Java的开发环境	6
2.2.2 Java的运行环境—Java虚拟机	8
2.3 安装Java编译器及其工具	8
2.4 小结	9
第三章 Java程序的基本结构	10
3.1 编写第一个 Java程序	10
3.1.1 “HelloWorld” Application	10
3.1.2 简单的类定义	11
3.1.3 基本I/O	11
3.2 为 Java程序增加新功能	12
3.2.1 变量和数据类型	13
3.2.2 控制流语句	14
3.2.3 操作符	15
3.2.4 表达式	18
3.3 数组和串	19
3.3.1 数组	19
3.3.2 串String和StringBuffer	22
3.4 main()和异常	25
3.4.1 main()方法	25
3.4.2 异常	26
3.5 包和编译单元	27
3.5.1 定义一个编译单元的包	28
3.5.2 使用其它包中的类和接口	29
3.6 小结	29

第四章 对象、类和接口.....	31
4.1 面向对象程序设计概念	31
4.2 对象	31
4.2.1 创建对象	32
4.2.2 使用对象	33
4.2.3 清除对象	34
4.3 类	35
4.3.1 类的声明	35
4.3.2 类的主体部分	36
4.3.3 声明成员变量	36
4.3.4 方法	37
4.3.5 类的成员的存取限制（存取修饰符的使用）	39
4.3.6 实例和类成员	40
4.3.7 构造和析构	43
4.4 子类、超类和继承	43
4.4.1 子类、超类、继承的概念	44
4.4.2 如何创建子类	44
4.4.3 Final类和方法	45
4.4.4 abstract类和方法	46
4.5 接口	47
4.5.1 什么是接口	47
4.5.2 定义接口	47
4.5.3 实现接口	48
4.5.4 接口作为类型	49
4.6 小结	49
第五章 API介绍	50
5.1 Java包	50
5.2 层次关系	51
5.3 小结	60
第六章 用AWT设计图形用户界面	62
6.1 AWT部件	64
6.1.1 部件的使用规则	65
6.1.2 部件的用法	66
6.1.3 版面管理器	102
6.1.4 事件控制	111
6.2 绘图和动画	114
6.2.1 AWT对绘图的支持	114

6.2.2	制作几何图形	115
6.2.3	设置和改变颜色	116
6.2.4	文本处理	117
6.2.5	图象处理	119
6.2.6	动画制作	123
6.3	小结	128
第七章	多线程	130
7.1	线程的概念	130
7.2	简单的线程例子	130
7.3	线程的属性	132
7.3.1	线程的主体	132
7.3.2	线程的状态	133
7.3.3	线程的优先权	135
7.3.4	服务线程	136
7.3.5	线程组	136
7.4	多线程的程序	138
7.4.1	公平、饥饿和死锁	139
7.4.2	同步	139
7.4.3	volatile变量	140
7.5	小结	140
第八章	输入 / 输出流	142
8.1	流	142
8.2	InputStream和OutputStream	142
8.2.1	InputStream	142
8.2.2	OutputStream	143
8.3	java.io中的类和接口	144
8.3.1	简单输入/输出流	144
8.3.2	过滤流	144
8.3.3	其它	145
8.3.4	java.io包中类和接口的继承关系	146
8.4	一个例子	147
8.5	小结	149
第九章	Applet	151
9.1	Applet的编写和执行	151
9.1.1	创建一个Java源文件	151
9.1.2	编译源文件	151
9.1.3	创建用于嵌入Applet的HTML文件	152

9.1.4 装载HTML文件.....	152
小结.....	153
9.2 Applet的生命周期及Applet中的主要方法.....	154
9.2.1 Applet的生命周期及Applet的基本框架.....	154
9.2.2 用于绘图和控制事件的方法.....	157
9.3 Applet的应用.....	158
9.3.1 Applet中的线程.....	158
9.3.2 用Applet制作图形用户界面.....	159
9.3.3 用Applet制作动画和声音.....	163
9.4 小结.....	172
附录A Java词法结构.....	174
附录B Java语言语法.....	178

第一章 Java 介绍

1.1 什么是 Java

Java 是一种软件技术，它是一种由美国 Sun Microsystems 公司为 Internet 而开发的电脑语言。Java 语言可以产生小应用程序（Applet），能够使网页（web page）由静态（static）转变为动态（dynamic），产生生动、活泼的画面。Java 是简化的 C++ 语言，使用者不需花费很多时间就可掌握。

Java 是一种简单的（simple）、面向对象的（object-oriented）、分布式的（distributed）、解释性的（interpreted）、健壮的（robust）、安全的（secure）、结构中立的（architecture neutral）、可移植的（portable）、高性能的（high performance）、多线程的（multithreaded）、动态的（dynamic）程序设计语言。

Java 突破了用户端机器环境和 CPU 结构，是一种“写一次即可在任何机器上执行（write once run anywhere）”的语言。Java 是有史以来第一套允许使用者将应用程序（Applications）透过 Internet 从远端的服务器（remote server）传输到本地的机器上（local machine）并予以执行的语言。应用程序提供者不需要知道使用者的电脑硬件（如 Sun、Intel 或 MAC 等）与软件（如 SW-UNIX、MACOS、Windows 或 NT 等）环境。

另外，Java 将安全性（security）列为第一优先考虑，具有杜绝电脑病毒传播的功能。

1.2 Java 对 Internet 的影响和作用

作为一种新的编程语言，Java 之所以流行，很大一部分原因在于它与 Internet 尤其是 WWW 的成功结合。Java 的目标是试图成为在 WWW 上传输动态可执行内容的“世界语”，但是，Java 给 Internet 带来的，远远不只是当用户链接到某一 WWW 页面时，可下载到客户端动态执行的活动画面，它的潜在价值是相当广泛的。总的来讲，Java 的作用包括：

- 开发适合于 Internet 环境运行的应用程序，或对现有应用程序进行升级。例如，用 Java 来编写数据库管理系统，使之易于在网络上使用。
- 成为新版本虚拟现实模型语言（VRML）的主要引擎。
- 编写 KbowBot（一种在 Internet 广泛漫游并负责搜集信息的程序），即智能代

理程序。

- 完全面向对象的编程方法学引入 Internet。

1.3 Java 的特点

作为一种新的编程语言，Java 具有如下的特点：

1. 简单性

这是 Java 最重要的特点。Java 语言的设计者虽然是基于 C++ 来设计 Java 的，但是却删去了 C++ 中几乎不用或很难用好的一些面向对象的特征，例如，Java 不支持 struct、union 和 pointer 数据类型；不支持 typedef 和 #define；不支持算符的重载和多继承性。Java 内部有一个自动分配和释放内存的系统，使程序员从繁琐的事物中解放出来，在编程中少犯错误。

2. 面向对象

面向对象是一项很强的设计功能，因为它促成明确的界面定义，并允许开发者建立可重复使用的“软件 IC”。简单地说，面向对象设计是一种以对象及其界面为重心的程序设计方法。Java 的面向对象机制实质上就是 C++，并包含 ObjectiveC 的一些延伸，以提供更强有力的解决方案。

3. 分布性

Java 是专为网络设计的，它有一个大类库用于支持 Internet TCP / IP 协议，如 HTTP 和 FTP 的通信。Java 代码经由 URL 操作资源就如同用 C 或 C++ 访问当地文件一样容易。

4. 可解释性

正如前面所提到的，由于 Java 特殊的编译、执行方式，编译生成的字节代码 (Bytecode) 可在任何能运行支持 Java 浏览器的机器上运行，Java 程序可独立于平台。因此 Java 的可移植性也很好。

5. 健壮性

Java 的类型法则比 C 和 C++ 严格，它不支持指针运算而只能用数组，这就排除了程序访问任意内存而可能产生的隐患。

6. 安全性

Java 的安全机制分为多级, 包括 Java 语言本身的安全性设计以及严格的编译检查、运行检查和网络接口级的安全检查。在语言方面, Java 摒弃了指针类型和数据类型的隐式转换, 从而对内存访问进行了严格的限制。在编译期间, Java 编译器并不分配内存, 而是推迟到运行时由解释器决定。这样, 编程人员就无法通过指针来非法访问内存。在运行期间, Java 的运行环境提供了四级安全保障机制: 字节码校验器、类装载器、运行时内存布局、文件访问限制。在网络接口级, 用户可按自己的需要来设置网络访问权限。另外, Java 的未来版本将采用公开密钥法以及其它加密技术, 来核实从网络上传输过来的代码的源主机及该代码的完整性。

7. 结构中立性

Java 的设计目标是要支持网络应用程序。一般而言, 网络是由许多不同的系统构成, 包括各种 CPU 与操作系统结构。为了让 Java 应用程序能够在网络上任何地方执行, 其编译器会产出一种具备结构中立性的对象文件格式, 编译后的程序代码可以在提供 Java 运行时系统的多种不同处理器上面执行。这不仅对网络应用很有帮助, 而且也很适合软件流通。Java 编译器是由产生与电脑结构无关的字节码指令来达到上述功能的。它们能轻易地在任何机器上解释, 并且动态地转换成机器码。

8. 可移植性

结构中立性是确保程序可移植的最重要部份, 不过除此之外还须很多条件配合。Java 指定基本对象类型的大小及其算术运算符的执行行为。库函数属于系统的一部份, 定义了一些可移植的界面。Java 本身具备相当好的移植性, 编译器由 Java 写成, 而运行时系统使用 ANSI C, 并有一个与 POSIX 相容的清晰的移植接口。

9. 高性能

虽然字节码性能已相当不错, 不过有些情形下还是要求程序有更高的执行性能。字节码可以在运行时由执行应用程序的特定 CPU 翻译成机器码, 这对于习惯使用一般编译器与动态装载器 (loader) 的设计者而言, 有点类似将最终的机器码生成器放到动态装载器之内。字节码格式在设计上即顾及机器代码的生成, 产生的程序代码有一定的合理性质: 它会自动配置暂存器, 而编译器在生成字节码时会执行一些最优化程序。

10. 多线程

多线程是一种应用程序设计方法。不幸的是, 要设计一个一次同时处理许多事件的程序, 比设计传统单线程的 C 与 C++ 程序复杂得多。

Java 拥有一组高复杂度的同部机制，它们以普及化的 C.A.R.Hoare 图为基础。将这些概念结合到语言中之后，它们即变得很容易使用且更为强韧。多线程的其它效益包括更好的交互式响应能力与即时执行行为。

11. 动态

从许多方面而言，Java 是一种比 C 或 C++ 更具动态特性的语言。它在设计上强调为运行中的运算环境提供支持。Java 是在稍后的阶段为模块与模块之间建立连接，并能更直接地运用面向对象设计体系。程序库可以自由地增加新方法和实例（instance）变量，而不会对它们的客户（clients）产生任何影响。

1.4 Application 和 Applet

Java 程序分为 Application 和 Applet 两种。两者都遵循 Java 语法。Application 是可以独立运行的应用程序，而 Applet 必须由 HTML 语言调用，不能独立运行。在 WWW 上，一般都使用 Applet，因为 Applet 规模很小，易于在网上传输。Application 较适合编写规模较大的应用程序，因为 Application 较为成熟，功能也很完善。

1.5 小 结

这一章中，简单介绍了关于 Java 的一些基本概念。现在，我们已经大致了解了什么是 Java，Java 的作用，Java 的特点，以及 Java 语言的两种形式：Application 和 Applet。

要点：

- 什么是 Java
- Java 对 Internet 的影响和作用
- Java 的特点
- Application 和 Applet 的异同

第二章 Java 的运行机制及其环境

2.1 Java 的运行机制

“Java”实际上包含两方面的含义：从狭义角度讲，Java 只是由 Sun 公司开发的一种编程语言，该语言可以创建一种可通过网络发布、动态执行的二进制“内容”（content）。广义的 Java 除 Java 语言外，还包括了一个客户机/服务器模式下的开发和执行环境。服务器端是一个完整的 Java 语言开发环境，而客户端是支持 Java 的 WWW 浏览器。Java 虚拟机（JVM）是可运行 Java 代码的假想的计算机。只要根据 Java 虚拟机的规格进行描述，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 代码都能够在该系统上运行。

源文件的编译、下载、解释和执行

Java 应用程序的开发周期包括编译、下载、解释和执行几个部分。

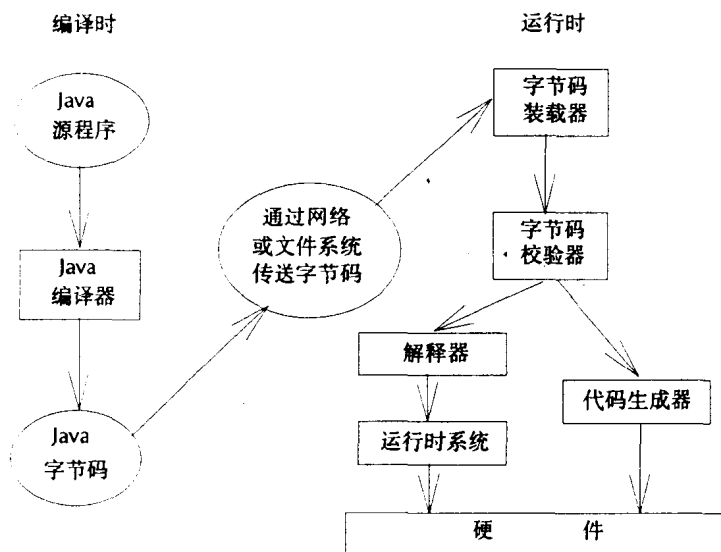


图 2.1 源程序的编译、下载、解释和执行

图 2.1 给出了从 Java 源代码开始, 经过 Java 编译器到字节码校验器, 最后到 Java 解释器的数据流和控制流。其中 Java 字节码装载器和字节码校验器并不限定字节码的最初来源, 代码可能来自本地系统, 也可能来自网络上的任意主机。

Java 编译程序将 Java 源程序翻译为 Java 虚拟机的可执行代码(字节码)。这一编译过程同 C / C++有些不同。当 C 编译器编译生成一个对象的代码时该代码是为在某一特定硬件平台运行而产生的。因此, 在编译过程中, 编译程序通过查表将所有对符号的引用转换为特定的内存偏移量, 以保证程序可自己运行。Java 编译器却不将对变量和方法的引用编译为数值引用, 也不确定程序执行过程中的内存布局, 而是将这些符号引用信息保留在字节码中, 由解释器在运行过程中创立内存布局, 然后再通过查表来确定一个方法所在的地址。这样就有效地保证了 Java 的可移植性和安全性。

2.2 Java 的开发运行环境

2.2.1 Java 的开发环境

Java 的开发环境分为基于 MS-DOS 或 UNIX 的 shelltool 的命令行开发工具和基于 Windows 的开发工具。命令行开发工具较为成熟, 而基于 Windows 的开发工具还正在开发和完善之中。

基于 MS-DOS 或 UNIX(S) 的 shelltool 的开发工具

目前, 服务器一端的 Java 环境包括如下开发工具:

- **Javac**: 即 Java 编译器, 用于把 Java 源程序编译成字节码。
- **Java**: 实际上就是 Java 解释器, 用于解释执行 Java 字节码, 这种解释执行不需要 WWW 浏览器的参与。
- **JavaDisassembler**: Java 分解程序用于分解 Java 字节码文件, 并返回有关 Java 程序的成员变量及方法等信息。
- **Javaprof**: 通过本命令, 用户可以了解 Java 程序在运行过程都调用了哪些资源, 包括类和方法的调用次数和时间, 以及各数据类型的内存使用情况等。
- **Javah**: 利用本命令可以在 Java 类中调用 C++代码。
- **JavaAppletViewer**: JavaApplet 浏览器, 用于测试和运行 Applet。
- **JavaDebuggerAPI** 和 **PortotypeDebugger**: 一个测试工具 API 和一个调试工具原形。
- **Java 类库**: 和任何一种面向对象的语言一样, Java 的开发环境也提供了丰富的

类库, 包括 I / O 类库、用户界面类库和网络类库等。

Java 开发工具有基于各种硬件平台的版本。目前 Sun 公司已经提供了在 Solaris2.3、2.4、2.5、WindowsNT、Windows95 和 MacOS7.5 上的开发工具, 一些第三方厂商也已经或即将提供 Windows3.1、Amiga、NeXT、SiliconGraphics 和 Linux 等操作平台上的 Java 开发工具。

基于 Windows 的开发工具

Sun 公司提供了基于图形方式的可视的开发工具 Java WorkShop。Java WorkShop 既是 Java 程序的开发工具, 同时也是一个 WWW 浏览器。图 2.2 是 Java WorkShop Release 5.0 版的界面:

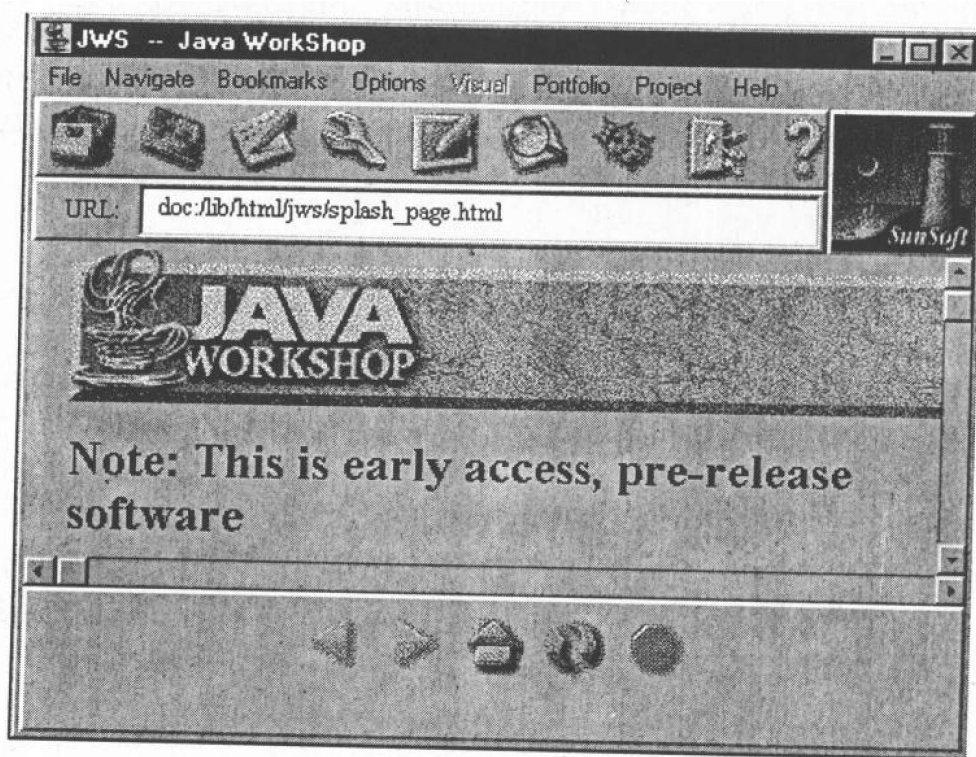


图 2.2 Java WorkShop Release5.0 界面

正如界面中的图标所示, Java WorkShop 有如下的工具:

- Portfolio Manager (公文管理): 可以创建一个 project, 从一个 Portfolio Manager

向另一个 Portfolio Manager 拷贝 project，并组织相关的 project 文件。

- **Project Manager**：可阅读和编辑一个 Project 的属性。属性定义了 project 的独立的特性，如 project 在 Web 页面上如何显示。

- **Source Editor**（源文件编辑器）：阅读和编辑 project 源文件。它也提供一些功能，可实现一些最常用的调试功能，如设置中断点、单步执行源代码。

- **Build Manager**：编译并连接（build）所有的 project 文件，或只编译并连接新修改过的一部分 project。若干 project 中有错误，Build Manager 会报错，并在源文件中指出错误。

- **Source Browser**：浏览组成 project 的类、方法和源文件的文档。也可在源文件中进行字符串的查找。

- **Debugger**：可加载并改正阻碍程序正确运行的错误。用 Debugger，可以启动、运行、单步执行和停止一个线程。也可以设置断点、条件表达式、捕获并忽略异常，以及检查程序中的包和类。

- **Project Tester**：运行 project。如果 project 是一个 Applet 或远程 Applet，Project Tester 会为它生成一个 HTML 文件，并在 Web 浏览器中运行这个文件；如果 project 是一个 Application，则 Project Tester 在 Web 浏览器之外运行它。我们无法运行一个 Java 包或图象 project。

- **Online Help**：提供 Java WorkShop 的线上帮助信息。

2.2.2 Java 的运行环境—Java 虚拟机

Java 虚拟机的设计目标是提供一个基于抽象规格描述的计算机模型，为解释程序的开发人员提供很好的灵活性，同时也确保了 Java 代码可在符合该规范的任何系统上运行。Java 虚拟机对其实现的某些方面给出了具体的定义，特别是对 Java 可执行代码即字节码格式给出了明确的规格。这一规格包括操作数的语法和数值，标识符的数值表示方式以及 Java 类文件中的 Java 对象和常量缓冲池在 Java 虚拟机中的存储映象。这些定义为 Java 虚拟机解释器开发人员提供了所需的信息和开放的开发环境。Java 的设计者希望给开发人员随心所欲使用 Java 的自由。

2.3 安装 Java 编译器及其工具

虽然 Java 的开发环境有命令行和窗口两种形式，但为了方便起见，建议你安装 JDK（The Java Developers Kit）。目前的版本是 JDK1.0.2，可用的平台有：SPARC Solaris、x86 Solaris、Windows NT、Windows95 和 MacOS。

我们可以通过 Internet 在 WWW 上下载 JDK1.0.2。只要你进入下面的地址，就可

以获得下载 JDK1.0.2 的信息:

http://java.sun.com/products/JDK/1.0.2/ftp_docs.html.

下载得到的是一个后缀为“.t.z”的压缩文件,我们用“gzip 文件名.t.z”命令和“tar -vxf 文件名.tar”将其解压,然后再运行 setup 文件便可生成一系列的子目录。这样,Java 的开发运行环境便安装好了。

目前可以从 Internet 上购买 Java WorkShop1.0,也可以从 Internet 上下载 Java WorkShop Release5.0 版。只要你进入下面的地址,便可得到相关信息:

http://www.sun.com/sunsoft/Developer_products/java/tnb/index.html.

2.4 小 结

这一章中,我们学习了 Java 的运行机制,Java 的运行开发环境,以及 Java 编译器及其工具的安装。我们已经掌握了学习 Java 语言的一些必要的背景知识和开发 Java 程序所必须的工具。

要点:

- Java 的运行机制
- Java 的开发、运行环境
- Java 编译器及其工具的安装