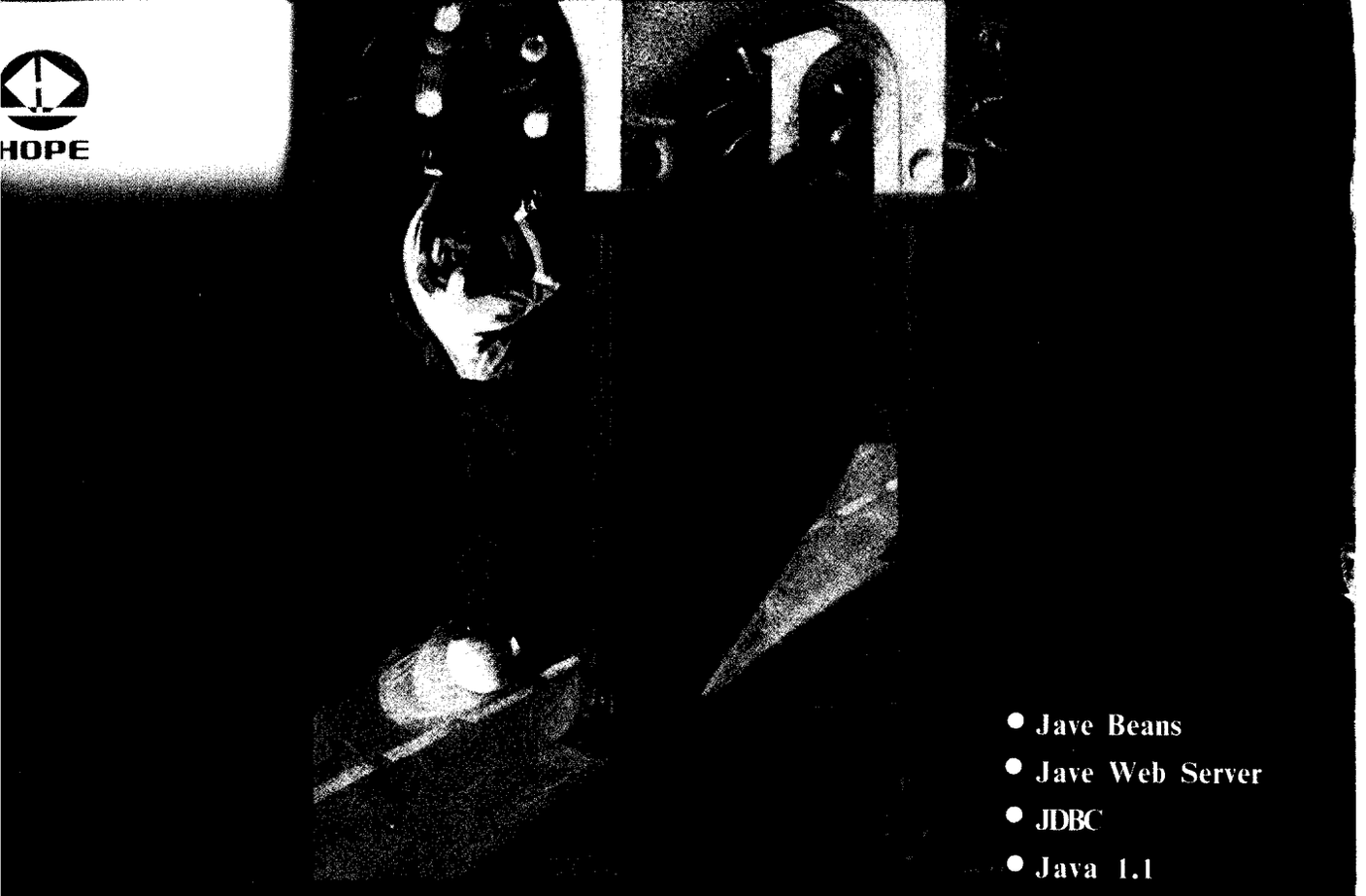

- Jave Beans
- Jave Web Server
- JDBC
- Java 1.1

精通JAVA

〔美〕K.S.西扬 J.韦弗 著
希望图书创作室 译

学报出版社
西蒙与舒斯特国际出版公司

Karanjit S. Siyan, James L. Weaver: Inside Java

Authorized translation from the English language edition published by New Riders Publishing.

Copyright © 1997 by New Riders Publishing.

All right reserved. For sale in Mainland China only.

本书中文简体字版由宇航出版社和美国西蒙与舒斯特国际出版公司合作出版，未经出版者书面许可，本书的任何部分不得以任何方式复制或抄袭。

本书封面贴有 Prentice Hall 防伪标签，无标签者不得销售。

版权所有，翻印必究。

图书在版编目(CIP)数据

精通 Java/(美)西扬(Siyan, K. S.),(美)韦弗(Weaver, J.)著；希望图书创作室译. —北京：宇航出版社，1998.5

书名原文：Inside Java

ISBN 7-80144-017-X

Ⅰ. J… Ⅱ. ①西… ②韦… ③希… Ⅲ. Java 语言-基本知识，IV. TP312Ja

中国版本图书馆 CIP 数据核字（97）第 23975 号

宇航出版社出版发行

北京市和平里滨河路 1 号(100013)

发行部地址：北京阜成路 8 号(100830)

北京东升印刷厂印刷

新华书店经销

1998 年 5 月第 1 版　　1998 年 5 月第 1 次印刷

开本：787×1092 1/16　　印张：38.25　　字数：898 千字

印数：1～7000 册　　定价：55.00 元

作者简介

Karanjit S. Siyan 是 Kinetics 公司的总裁，他曾主办了许多国际研讨班，内容涉及 Solaris&SunOS、TCP/IP 网络、PC 网络集成、Windows NT、Novell 网络以及使用模糊逻辑的专家系统。他还在美国、加拿大、欧洲以及远东地区开办高技术研讨班。Siyan 博士曾在《DR. Dobbs Journal》、《The CUsers Journal》、《Databasd Advisor》等杂志上发表文章，积极参与 Internet 的研究工作。多年来 Karanjit 一直从事计算机语言的工作，如 C/C++，Pascal，LISP，Algol，Ada，MSL，CMS-2，Jovial、Java 等，并为许多语言编写了编译器。Karanjit 获得计算机科学博士学位。他曾在 ROLM 公司担任技术支持部的主管，并是许多项目的技术管理员和软件开发者。在担任顾问工作期间，Karanjit 为客户编写了许多编译器和操作系统开发工具。他的兴趣包括基于 UNIX、Netware Windows NT 以及 OS/2 的网络。他还积极参与许多计算科学学科的工作，如网络、操作系统、编程语言、数据库、专家系统以及计算机安全性。Siyan 博士已获取 Windows NT 技术认证以及 Novell 网络的 Master CNE 和 ECNE 的认证。他写了许多书。现在，Kranjit Siyan 及其妻子住在美国蒙大拿州。

Jim Weaver(jlweaer @ netusal. net)在过去的 15 年中，作为顾问和 EDS 的高级系统工程师，开发了许多商业、制造业以及工程应用程序。目前，Jim 从事 Internet 的研究工作，主要研究如何使 Internet 成为一个执行平台。他还开始研究漫游 Internet 的 Java 代理程序。他喜爱弹吉它。Jim 和他的妻子有两个孩子，他们住在印第安纳州。

25

鸣　谢

Karanjit S. Siyan 的致谢词

感谢那些在编书的过程中给予了很大帮助的人们。衷心感谢我的妻子 Dei 对我的关心和支持。感谢我的父亲 Ahal Singh，母亲 Tejinder，我的兄弟 Harjeet 和 Jagjit；我的姐妹 Kookie 和 Dolly。还要感谢 Margaret Cooper Scott，Cathryn 和 Bob Foley，Craig 和 Lydia Cooper，Robert 和 Janie Cooper，Heidi 和 Steve Bynum，Barbara 和 Edward L. Scott 和 Jacquelyn Mcregor，感谢他们的关心和大力支持。特别要感谢 Mother，Saint Germain，EL Morya Khan，Bhaywan Krishna 和 Babyji。没有他们的鼓励，就不可能有这本书。

我要感谢 Bob Sanregret 和 Anders Amundson，最初是他们引发我写计算机教材的兴趣。我还要衷心感谢 Learning Tree 的许多人，感谢他们的帮助和对许多项目的支持。尤其要感谢 John Moriarty，Rick Adamson，Dr. David Collins，Eric Garen，Matri Sanregret，Richard Beaument，Mark Drew，David O'Neil 和 Robin Johnston。Boyd Badten 为本书中的动画应用程序创建了图像文件，谢谢 Boyd。

我衷心感谢 Marta Partington 和 Naomi Goldman，感谢他们对本书所给予的大力支持。特别要感谢 Don Fowley，Jim LeValley，Sean Angus，Gina Brown，Nancy Maragioglio 和 Krista Hansing 对本书的帮助。最后感谢 Drew 和 Blythe Heywood 多年来给我的友谊、支持和鼓励。

Jim Weaver 的致谢词

感谢 New Riders 出版公司员工的辛勤工作，终于使本书出版面世了。尤其要感谢 Sean 和 Nancy 使我成为本书的合著者之一。感谢 Gina Brown，Naomi Goldman，Krista Hansing 和 Jim Mathis 对本书的编辑、排版。还要感谢 Colin 把我推荐给 New Riders 出版公司。

感谢 Julie Weaver，Lori Weaver，Kelli Weaver，Marilyn Prater，Walter Weaver 和 Ken Prater，谢谢他们的关心和大力支持。

最后向 Java Soft 项目组的老成员和新成员表示祝贺和感谢，是他们造就了信息技术行业划时代的革命，并使 Java 语言充满情趣。

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第 1 章　Java 简介

Java 是 Sun Microsystems 公司开发的一种新的编程语言。Java 有许多特点使得它非常适合于 Internet 应用程序，如 WWW，虽然 Java 并不仅限于 Internet。Java 跨平台可移植性使得它非常有吸引力，是许多内部协作网的理想选择。已有人使用 Java 开发出了大规模的软件系统，这充分说明 Java 可以作为一个通用的工具来编写商业和科学逻辑运算等应用程序。

丰富的类和对象继承使得程序员可以重用已有的程序代码，并快速地编写复杂的应用程序。

1.1　Java 语言

Java 语言的诞生可以追溯到软件工程师开发可移植的家用电器控制软件上，如烤箱、面包炉、TV、VCR、电灯、BP 机、电视顶置盒、个人数字助理(PDA)等等。1991 年 4 月，Sun Microsystems 公司的一组人员开始了一个代号为“Green”的工程。Green 工程的目标是开发出一个用于开发家用电器控制逻辑的系统。Sun Microsystems 公司的员工很快意识到，在家用电器中使用的处理器没有统一的标准。为了简化开发工作，他们需要一个与平台无关的开发环境。

Sun Microsystems 公司的 James Gosling(UNIX EMACS 编译器和 NeWS window 系统的创造者)起初试图扩展 C++语言。这需要花费很多的精力，并且不会得到最好的效果。因此，Green 工程就着手开发一种称为 Oak 的语言。James Gosling 将其命名为 Oak 是因为在他设计该语言的结构时，窗外的一棵 Oak 树映入了他的眼帘。后来，Oak 这个名字不得不放弃，因为较早的一种语言也命名为 Oak。经过很长一段时间的构思后，在小组成员去附近的一家咖啡店时，灵感就来了。因此，与流行的说法相反，Java 不是“Just Another Vague Acronym”的缩写。Oak 开发环境有四个要素：Oak 语言、称为 GreenOS 的操作系统、用户界面和一个称为 * 7 的类似于 PDA 的设备。“ * 7”取自一个电话序号，在 Green 工程项目组办公室所在的大楼 Sand Hill 中，采用“ * 7”这个电话序号来接听电话。 * 7 设备被设计为一个通用控制设备，家用电器制造商可以将其加入到自己的产品中，其用户界面包括一个栩栩如生的全彩色家居或办公室示意图，使用者可以通过触摸屏来进行操作。

1993 年，Green 工程小组加入到 Sun Microsystems 的一个下属公司 FirstPerson，投标于 Time-Warner 公司的一个电视研究项目。然而，到 1993 年 6 月，Time-Warner 公司采用了 Silicon Graphics 公司的建议，Green 小组的投标失败。1994 年，另一笔与 3DO 公司的生意也失败了，寻找新的商业伙伴前途渺茫。FirstPerson 被取消，刚组建的公司被解散。

Green 工程小组的一半成员去了 Sun Interactive，研究数字视频服务器；另一半成员回到 Sun 公司研究多媒体和网络计算。

正当 Green 工程小组遭受这一切挫折时，Web 获得了很大的进展，尤其是能够给用户提供相当灵活性和导航能力的 GUI Web 用户软件的开发工作获得了长足的发展。那时，World Wide Web 已主宰了 Internet 的发展。Sun Microsystems 意识到了 Oak 技术的潜能，开发了一

个 WebRunner 浏览器，后来改名为 HotJava。

*7 设备和 VOD(Video On Demand)技术的开发工作使得 Java 语言成熟起来。Java 编译器最初由 C 语言写成，后由 Arthur Van Hoff 用 Java 改写。编译器是一种复杂的软件，用 Java 语言改写了 Java 编译器表明 Java 语言本质上是一种通用的编程语言。

1995 年 5 月 23 日，Sun Microsystems 在 Sun World'95 上发布了 Java。Java 的趣味性和对将来网络计算的承诺迅速引发了一场 Java 热。流行的浏览器如 Netscape Nevigator 和 Internet Explorer 都融入了 Java 技术。这些浏览器可以运行从远程服务器上下载的 Java 程序。为 Web 开发的 Java 程序称为 Java 小应用程序(Java applet)。Java 小应用程序以 URL 地址的形式嵌入到 HTML 页面中。因为 URL 地址可以引用网络上的任何计算机，所以可以使用 HTTP 协议和 HTML 语言将执行代码分散放置于 intranet 或 Internet 上。

1.1.1 特征

Sun 将 Java 语言描述为一种简单的、分布式的、解释执行的、安全的、结构中立的、可移植的、高性能的、多线程的、动态的语言。这有许多的形容词，有些人甚至称其为蜂鸣词(意思是说象蜂鸣器似的一连串说出一堆的形容词)。其中的一些蜂鸣词也用于描述其他语言。但 Java 语言的独到之处在于，它是第一种既可以编写普通的应用程序也可以编写专用于 Internet 和 intranet 应用程序的语言。

在 Internet 和 intranet 应用中，Java 程序通常是在 Web 浏览器中查看 Web 页面时运行。在 Web 浏览器中运行的 Java 程序称小应用程序(applet)，可以在 Web 浏览器之外独立运行的 Java 程序称为 Java 应用程序(application)。本书中你将学习这二者之间的区别。

1.1.2 使用简单

Java 在设计时就注意保持其简单性，同时又要有足够强大的功能以便完成 Web 中的网络计算任务。这种易用性和强大的功能也有助于开发 intranet 应用程序。为了达到简单性，Java 语言的设计人员精简了语言构件。这样，既方便了人们学习，同时也使得编译器很小且容易实现。

设计人员采用了 C/C++的语法规则。C/C++程序员非常多，他们会发现很容易转到 Java 语言，Java 语言很容易学习和使用。C/C++语言的某些特征被删除，以便保证 Java 的易用性和安全性。例如，Java 语言不支持 goto 语句；相反，它提供异常处理，采用 break，continue 和 finally 语句。C/C++语言中的头文件也被删除。Java 中没有 #include 预处理语句；相反，使用 import 语句有选择性地引入某个包中的 Java 类或所有的 Java 类。

为了支持大软件的模块化开发模式，采用了包(package)的概念。包是一组功能相似的 Java 代码集合体。与 C/C++语言的其他不同之处包括：删除了某些数据结构，如 struct 和 union。相似的概念可由类的构造来实现。为了保证 Java 语言的简单性，C++中的运算符重载和多重继承也被删除了。

与 C/C++语言的最大区别是删除了对指针的直接引用。虽然指针在 C/C++中功能非常强大，但这需要一套规则来保证正确地使用它们以便代码中不会存在错误。而维护 C/C++代码的实际经验告诉我们，C/C++中指针的使用容易导致错误。Java 自动处理对语言对象的引用和释放。这使得你不必将精力花在引用指针、错误引用指针、内存释放等问题上。对象是

动态创建的，并且 Java 后台环境自动对垃圾进行回收。在 C/C++程序中，不正确的内存分配也是常见的错误之一，而在 Java 环境中，这些是自动完成的，从而减少了内存方面的错误。

虽然 Java 语言同 C/C++相比已简化不少，但它提供了许多预定义的类以完成 I/O、网络和图形操作。这使得 Java 语言容易使用同时功能又足够强大，适于开发基于网络的 intranet 应用程序。

1.1.3　面向对象的本质

Java 是一种面向对象的语言。在这种语言中，所有的程序和数据都存在于对象中。对象就是数据以及对数据进行操作的程序的集合。这些程序是专为对象内的数据而编写的，称为对象的方法(method)。事实上，方法的一个参数就是对象本身(在 Java 语言中以“this”这个保留关键字来表示)。正如“method”这个名字所示，对象的方法是对象内数据进行操作的机制。数据和方法描述了对象的本质。

对象在 Java 语言中是由类来构建的。类在 Java 中的使用非常基本，不可能不使用类而编写出一个有意义的 Java 应用程序。

通过继承机制，Java 也支持代码重用。一个类可以由另外的类派生。这个过程称为继承(inheritance)或子类(subclassing)。Java 语言提供了一个非常有用的类体系结构。最顶层是 Object 类，这是所有其他类得以派生的根类。缺省时，一个新创建的类总是继承 Object 类，即使你并没有明确定义所继承的父类。预定义的 Java 类体系使得 Java 环境丰富多彩。

事实上，学习 Java 的难点不是语法或语义。这两者都比较直观，容易学习。难点在于熟悉预定义的类体系和方法以及方法的目的。当然，你可以自己构建一套类体系来取代已有的类体系，但效率不如已有的高。通过继承，你可以重用代码，并且派生现有的类体系结构。当所需的类体系结构能力不足时，你可以创建自己的类体系。

1.1.4　分布式语言

Java 内在地支持网络应用程序。这个网络可以是 Internet 或 intranet。Java 通过预定义的包 java.net 来提供网络能力。这个包提供了许多类以简化网络上不同计算机运行的应用程序之间的网络通信。使用 Java，你可以同样的方式访问远程和本地文件。并且，Java 语言支持虚电路网络连接，可用于建立 intranet 上的分布式客户/服务器应用程序。

1.1.5　解释语言

Java 编译器并不生成可执行程序的机器语言指令。相反，Java 编译器生成一种中介代码，称为字节码。Java 解释器读取 Java 字节码，并在内部使用一种抽象机模型来执行该字节码。Java 解释器和这种抽象机模型称为 Java 虚拟机 (JVM, Java Virtual Machine)。

Java 字节码是一种与硬件结构无关的表示程序的方法。因为 Java 程序是由 Java 字节码解释执行的，所以 Java 语言是一种解释性的语言。在解释环境中，将多个目标模块组合成二进制可执行机器指令集的连接过程没有了，连接阶段由 Java 环境加载类所取代，解释性环境如 Java 支持快速原型和程序开发。

如果应用程序不能容忍 Java 字节码解释执行的开销，可以采用 JIT(Just In Time)编译器。

JIT 编译器将 Java 字节码编译为程序所需要运行的计算机机器语言指令，这极大地提高了程序的性能。当然，JIT 编译器的输出结果不再是与硬件平台无关的了。

1.1.6 强健的语言

程序员谈论语言的强健性时是指该语言在编译和运行时支持消除易于出错的结构。Java 是一种严格定义的语言，对如何使用对象有明确的定义。例如，不可能在不损失精度的情况下将一个浮点数赋给一个整数变量。在编译阶段，编译器会捕捉这样的错误和其他类似的错误。通过强制类型转换或编写转换程序，你可以将某种类型的对象转换成其他类型，但这种转换要程序员来明确指明，Java 语言不会自动完成这种事。

Java 内存模型自动进行垃圾回收，这就减少了类的运行错误。不支持对指针的直接使用，这就消除了与指针有关的运行错误，如偶然改写数据和内存崩溃等等。Java 解释器也执行运行时检测，例如，确保阵列和字符串操作在阵列范围内进行等等。

Java 语言支持显式异常处理，使得程序员可以编写出强健的程序。除了预定义的异常状态外，程序员也可以定义自己的异常状态。

1.1.7 安全的语言

因为 Java 语言希望在网络环境中运行，而网络上各主机的情况是不确定的，因此，Java 语言的安全性是一个重要的设计目标。Java 代码放在 Web 服务器上，可以由 Web 浏览器提供的 JVM 下载并运行。没有办法阻止病毒和其他恶意程序冒充合法代码并给客户机和网上其他计算机带来损失。

许多病毒程序常使用的一个技巧是通过巧妙地运用地址变量如指针来获取计算机的资源，这就是为什么 Java 语言设计者决定放弃指针的重要原因之一，这也消除了安全方面的一个重要隐患。

类的内存分配和布局由 Java 环境透明地完成。因为 Java 程序员不访问内存布局，他不能确切地知道内存的实际布局，这使得病毒程序很难访问 Java 程序的内部数据结构。

在 Java 程序解释前，Java 运行系统进行字节码验证。字节码验证是一个必须进行的过程，采用了数学算法来保证程序不会违反系统的完整性。并且，从网上下载的程序分配在与本地类不同的名字空间中，这样可以防止恶意的 Java 小应用程序替换标准的 Java 类。

Java 可以阻止和避免导致程序失常的常用伎俩，然而，它并不是一个百分之百的安全环境。许多蓄意和狡猾的黑客正试图找到 Java 的弱点，如果这种事情发生，Sun Microsystems 将不得不弥补这些安全漏洞并在黑客之前采取行动。

虽然在 Internet 环境中安全性很重要，但在 intranet 环境中，安全性并不那么突出。主要原因是 intranet 环境中，恶意用户比 Internet 少得多。因为 Java 可在 Internet 上提供一个安全的运行环境，所以其当然可以满足大多数 intranet 的需要。

1.1.8 结构中立

Java 程序编译为字节码，由 JVM 解释执行。字节码与任何处理器类型和计算机结构无关。使用字节码可使 Java 代码在任何支持 JVM 和 Java 解释器的计算机上运行。

Java 语言的结构中立性对 intranet 很重要，它使得应用程序只需要用 Java 编写一遍。同样

此为试读，需要完整PDF请访问：www.ertongbook.com

的程序代码可以在不同的客户机或服务器平台上运行。图 1.1 显示了同一个 Java 小应用程序代码下载到不同的客户机平台上。

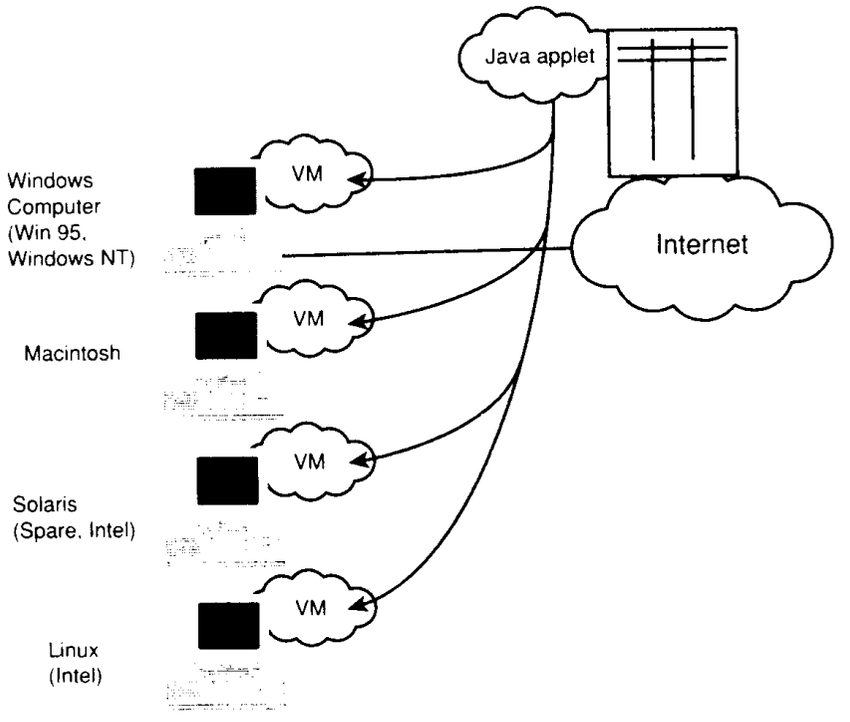

图 1.1　Java 代码的运行与硬件平台无关

目前，许多公司花费大量的人力、物力和财力，试图覆盖他们所需要支持的所有硬件平台。使用不同版本的 UNIX，Windows NT，Windows 95，OS/2 和 macintosh 计算机，就要开发可在这些不同平台上运行的软件，这是一个巨大的挑战。Java 在帮助解决这一问题上大大地前进了一步。

1.1.9　可移植的平台

因为 Java 代码结构中立，其可以在任何平台上运行。代码只需编写一次，字节码可以发布给不同的平台，不需要修改就可以运行。其他语言中常见的一个移植问题就是基本数据类型如整数、字符和浮点数的大小(比特数)不同。

例如，你需要知道基本的整数类型是 16 位还是 32 位。在 Java 语言中，所有基本类型大小都是一样的，不管其在哪种平台上运行。例如，Java 语言中的整数总是 32 位，不管其是在 UNIX 上运行还是在 OS/2 上运行。使用标准的整数定义可以避免错误，如上溢或下溢。而这些错误对于基本数据类型大小假定不一致时是经常出现的。因为 Java 小应用程序是基于 GUI 的，并且可以是多线程的，因此只有具有某些特征的计算机上才能运行 Java 程序。例如，在不支持 GUI 界面或多任务的 MS-DOS 计算机上难以运行 Java 代码。

1.1.10　高性能的工具

与其他描述性语言如 BASIC、Visual Basic，Shell Scripts 和 Perl 相比，Java 语言的性能很高。然而，Java 比 C 语言慢约 20 倍。对大多数交互式应用程序来说，Java 的速度足够了。

如果想让 Java 的速度与 C 相当，必须使用 JIT 编译器。许多开发商都提供 JIT 编译器，如

Sun，Borland，Symantec 等等。Web 浏览器准许 JIT 编译器提供加速器以提高 Java 小应用程序的性能。Java 字节码可以快速地翻译为机器指令。翻译为机器指令的字节码其性能与 C/C++程序相差无几。

1.1.11 多线程的语言

Java 是仅有的几种在语言本身就以多线程的形式支持多任务的语言之一(另外有 Ada 语言)。这意味着你可以编写有多个线程同时运行的 Java 程序。一个线程是程序中一条独立的执行流，程序中可以使用线程实现一定的并行计算。每个线程执行一特定的功能，使用几个线程可以并行进行一些活动。在单 CPU 计算机 上，每个线程轮流在 CPU 上运行，因此看起来线程都在同时运行。实际上，每个线程都运行一段时间，然后另外的线程运行。在多处理器计算机上，线程可以在不同的处理器上同时运行。为了避免各线程由于临界操作而互相死锁，你可以用同步关键字指明临界区域。大多数使用多任务或多线程的程序都对其所运行的操作系统进行低级系统服务调用。因为系统服务调用的不同，这些程序不能移植到其他的操作系统中去。然而，使用 Java，你可以编写一个多线程的程序，该程序可以在不同的计算机和操作系统之间移植而不必作任何修改。

1.1.12 动态的语言

Java 是一种动态的语言，意思是说它在需要时才加载相应的类。程序运行时，通过检查与类相关的运行类型信息，你可以确定一个对象属于哪个类。运行类的应用使得它可以容易地进行类的动态连接。

1.2 小　结

本章向你介绍 Java 语言的一些特点。Java 是 Sun Microsystems 推出的一种新的编程语言，该语言的许多特征使得它非常适合于 Internet 应用程序如 World Wide Web。本章未对 Java 特征作细致讨论，目的是让你了解 Java 语言可以做什么。关键要了解 Java 语言并不仅限于 Internet。Java 的跨平台可移植性非常有吸引力。对许多内部协作网来说，Java 是一个理想的选择。

第2章　Java入门

本章讨论什么是Java程序以及如何编写一个简单的Java程序。本章的目标是快速浏览一下Java语言的基本组件，而不是讲解过多的细节和该语言的强大功能。当你理解了如何编写Java程序的基本概念后，你就会轻松掌握后面几章中提到的复杂概念。本章并不假定你一定是一个专业程序员。如果你已经有了某些编程语言概念，如BASIC，Pascal，Ada或C/C++，那么，本章中提到的许多概念就当是复习好了。然而，本章将向你介绍一些Java的基本组件。在本章结束时，你应能够编写一些简单但有用的程序。为了达到这个目的，本章将集中于Java程序的某些基本概念，如变量和常量、控制流以及基本的输入/输出。其他的一些概念如面向对象编程和设计类的结构对于编写大的程序很重要，但这些概念都有意安排在其他章节学习。虽然这种方法可以让你理解如何编写简单的Java程序，但这种教学方法也有几个缺点。其中之一就是可能误导你对Java功能的理解，因为本章并不介绍Java的全部功能。你也应该清楚，某些例子是经过了精心挑选，因为本章并没有介绍Java的全部功能。另一个缺点是后面几章会不断地重复这些概念，但解释更深入些。

有多种方法来编译Java程序。有些是免费的，有些需要购买。本章中的所有例子都使用本书配套光盘(须另购)上的JDK(Java Developers Kit)命令行编译器；使用JDK提供的Java虚拟机来执行Java程序。商业化的Java编程工具有Symantec公司的Visual Cafe和Microsoft公司的Visual J++。

2.1　编写一个简单的Java程序

首先，确认你已经正确地安装了JDK或其他的Java编译器。在下面一节中，你将编写一个程序输出如下的语句：

```
Hello, World! I am a simple Java program!
```

为此，你需要学习以下几个部分：

- Java程序的基本结构
- Java语言中如何执行系统输出
- 如何创建和编辑本程序
- 如何编译本程序
- 如何执行本程序

基本上，对更大、更复杂的程序也使用相同的步骤。因此，当你理解了如何完成这些步骤来编写简单的Java程序后，你也就掌握了如何编写更大的Java程序。

2.1.1　创建、编译、运行一个简单的Java程序

下面是一个简单的Java程序：

```
class HelloWorld
```

```
{
    public static void main( String args[ ])
    {
        System.out.println("Hello, World! I am a simple Java program!") ;
    }
}
```

要使用本程序，你必须先将该程序存成文本文件。该文件必须以 .java 作为扩展名。目前发行的 Java 版本中，你必须以类名 HelloWorld 作为该文件的名字。类名是在上面这个程序中“class”之后的名字，描述程序的代码放在大括号{}之内，如下：

```
class classname
{
    ...
}
```

在前面的例子中，文件名必须是 HelloWorld.java，以便与类名 HelloWorld 相匹配。Java 中的类名与大小写有关。要编译和运行该程序，你必须将类名该为 helloworld。在 Java 语言中因为类名与大小写有关，所以下面的类是不同的：

```
class HelloWorld
{
    ...
}
class helloworld
{
    ...
}
class helloWorld
{
    ...
}
```

注意，因为 Java 文件的后缀是“.java”，你不能使用 MS-DOS 编辑器创建这些文件名。MS-DOS 文件名的后缀最多 3 个字符。

在 UNIX 下，文件名也区分大小写。然而，在 Windows NT 和 Windows 95 中，文件名不区分大小写，虽然文件名保留大小写。这意味着如果在一个有文件名为 HelloWorld.java 的目录下创建一个名为 helloworld.java 的文件，则会失败。说的再明白一点，就是在 Windows NT 和 Windows 95 环境下，如果 Java 程序都保存在同一目录下，则根据类名所生成的文件名应是唯一的。

要创建 Java 文本文件，可以使用文本编辑器创建一个名为 HelloWorld.java 的文件。你可以使用任何支持长文件名的文本编辑器来创建和保存该文件。

下一步就是编译这个程序。可以使用 javac 这个命令行编译器来编译该程序。打开一个命令提示符窗口，将目录转到保存 HelloWorld.java 文件的目录下。输入“javac HelloWorld.java”，输出如图 2.1 所示。

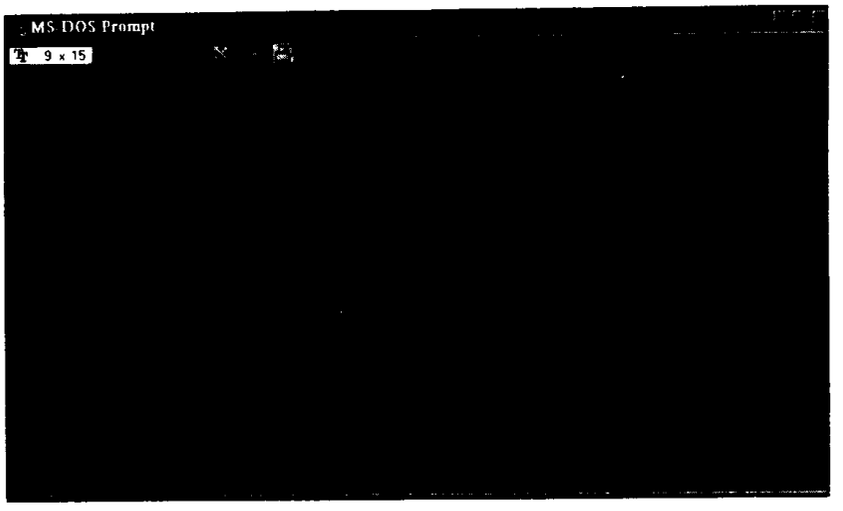

图 2.1　在命令行编译 HelloWorld 程序

要运行这个编译完的程序，将类名作为参数传递给 Java 虚拟机 java。图 2.2 显示了在命令行运行 HelloWorld 程序得到的输出结果。

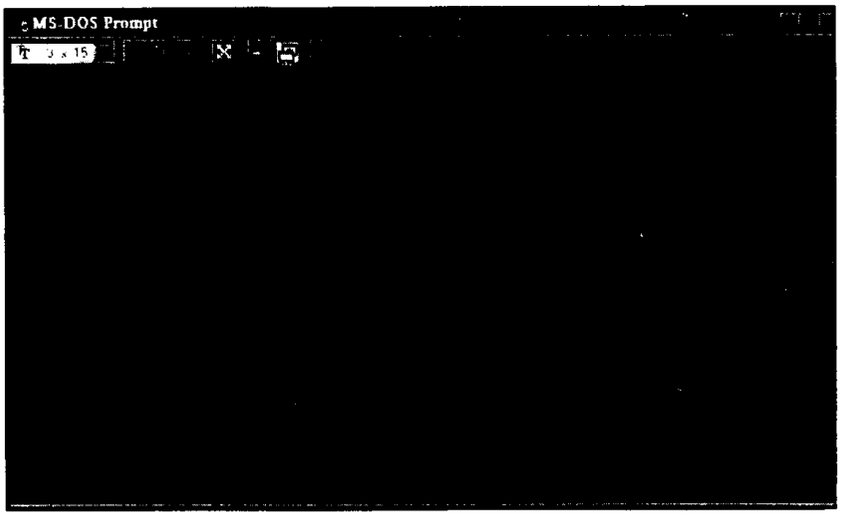

图 2.2　在命令行执行 HelloWorld 程序

2.1.2　理解 Java 程序的结构

HelloWorld 这个 Java 程序列出了一个 Java 程序的基本结构。

```
class HelloWorld
{
    public static void main(string args[] )
    {
        System.out.println("Hello, world! I am a simple Java program!");
    }
}
```

下面一节描述了这个简单的 Java 程序的结构和意思。有趣的是，要解释这个简单的 Java

类，你需要理解几个编程语言的概念和 Java 对象的概念。

(1)类的结构

注意，HelloWorld 程序的代码嵌入在如下的语法中：

```
class 类名
{
    程序的其他代码
}
```

在 Java 中，使用类来定义一块程序代码。以关键字 class 开头，后跟类名，然后是大括弧{}。程序代码放在大括弧{}中。在 Java 中，大括弧及其中的代码称为一个程序块(block)。

在 Java 语言中，保留关键字如 class 有特殊的含义，不能用于程序中定义数据变量和函数。在 HelloWorld. java 这个例子中的关键字还有 public，static 和 void。

所有包含程序代码的 Java 程序必须至少要有一个类。在这个类中是实际的数据和对数据进行操作的程序代码。在 Java 中，采用类的方法来描述对象的概念。

(2)简单的 Java 对象

一个对象是数据以及对数据进行操作的代码的集合体。图 2.3 显示了对象的概念表述。在这个图中，数据显示在对象的核心中。环绕数据的是对数据进行操作的程序。在对象中定义的对数据进行操作的程序称为方法。在其他的面向对象的语言中，如 Xerox Palo Alto 研究中心开发的 Small Talk，方法也称为消息，所以某些 Java 文档中也将对象中对数据进行操作的程序称为消息。

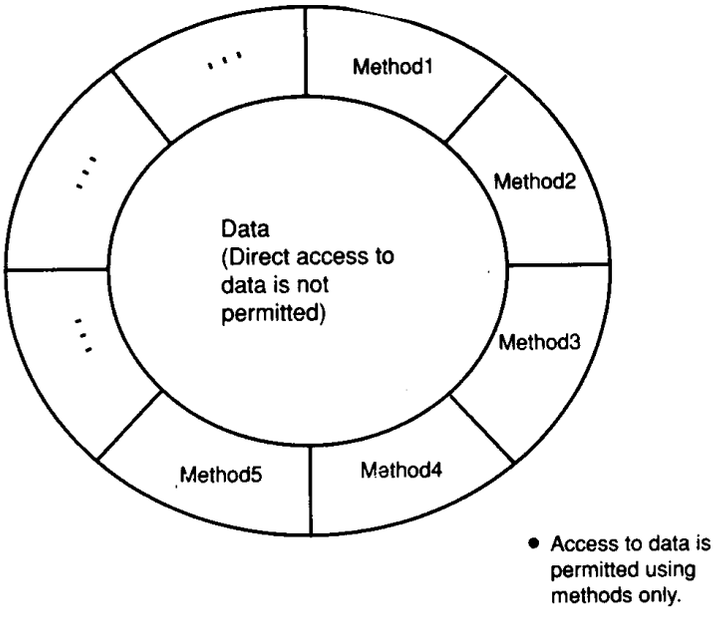

图 2.3 对象的概念表述

导致对象中的程序运行的过程称为激活一个方法、调用一个方法或给对象发送一个消息。从概念上说，就是当对象的方法被激活时，或当给对象发送一个消息时，程序代码就修改对象中代表对象状态的数据。

此为试读，需要完整PDF请访问：www.ertongbook.com