



董 慧 等编著

JAVA

JAVA语言系列丛书

实用教程



電子工業出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.co.cn>

Java 语言系列丛书

Java 实用教程

董慧 李旭晖 陆伟 编著
张俊 徐翰伟

电子工业出版社
Publishing House of Electronics Industry

内 容 简 介

Java 语言具有面向对象、高性能、可移植性、动态性、稳定性和安全性、分布式等优良特性, 为用户提供了良好的软件开发环境; 在 Internet 上 Java 是网络上的世界语, 它的成功为 Internet 和 WWW 开辟了一个崭新的时代。

本书共分十二章对 Java 语言的内容、功能、特性和实际运用作了深入浅出地、系统地、全面地介绍, 结构严谨、布局合理、重点突出、例程丰富, 能使读者很快地掌握 Java 程序设计的方法和技巧。

本书既可以作为高等院校计算机、通信、信息处理及相关专业的本科生和研究生学习 Java 语言的教材和教学参考书, 也可供其他工程技术人员, 特别是软件开发者自学使用。

351/02/58

丛 书 名: Java 语言系列丛书

书 名: Java 实用教程

编 著 者: 董慧 李旭晖 陆伟 张俊 徐翰伟

责任编辑: 张毅

排版制作: 电子工业出版社计算机排版室

印 刷 者: 北京农业印刷厂

出版发行: 电子工业出版社出版、发行 URL: <http://www.phei.co.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 19.25 字数: 500 千字

版 次: 1997 年 9 月第 1 版 1997 年 9 月第 1 次印刷

书 号: ISBN 7-5053-4318-1
TP·1969

定 价: 27.00 元

凡购买电子工业出版社的图书, 如有缺页、倒页、脱页者, 本社发行部负责调换
版权所有·翻印必究

前 言

1995年4月,我们承担了国家教委中国教育和科研计算机网示范工程应用系统专题——三峡地区和湖北旅游资源数据库的研究,根据项目要求,我们开始对Java语言与HTML语言进行探讨。当时,鉴于国内还没有多少资料可查,我们花了几个月时间研究了国外的许多专著。后来,我们武汉大学的校园网建成,在网上我们学到了许多丰富的Java信息资源,充实了我们写作本书的内容,也为我们完成课题打下了基础。在研究与学习过程中,我们深深地被Java吸引住了,它的风格、性能、特性正是我们梦寐以求的。过去,我们在开发MIS系统中,往往被多变的需求所困扰,而Java中的构件思想却很好地解决了MIS系统中长期无法解决的问题。在此,我们愿将此书献给支持与帮助过我们的同仁,献给亲爱的读者。

Java是美国Sun公司于1995年正式推出的一种全新的纯面向对象的编程语言,它具有高性能、可移植性、动态性、稳定性和安全性、分布式等优良特性,特别适合Internet/Intranet对编程的要求,又能和WWW很好地协同工作,人们称誉它为“网络上的世界语”。

Java可以认为是从C++派生出来的,它从C++中继承了大量优质的语言成份,但抛弃了C++中的冗余和容易引起问题的功能;增加了对多线程、异常处理、网络和多媒体编程等方面的支持。本书深入浅出的、系统地、全面地分析了Java面向对象、多线程、异常处理、网络和多媒体编程,以及Java与C、Java与HTML接口。大量的例程为读者学习提供了方便。

全书共分十二章。

第一章介绍了Java技术特性,应用前景;Java开发工具JDK以及HotJava。

第二章介绍了Java语言的基本类型、数组、字符串,各种数据类型及其间的相互转移、变量以及运算符和表达式。

第三章介绍了Java语言中与C相似的非面向对象的特性,内容包括Java与C++之间的差异,各种控制语句,如条件语句、循环语句、开关语句、转移语句等。

第四章介绍了Java的面向对象程序设计方法,其中包括:对象分类、接口、继承、包和对象容器的利用。

第五章介绍了Java的用户界面。其中包括:Java的用户事件处理、基本用户界面设计、高级用户界面设计,高级用户界面的设计。这些设计方法与思想是Java用户界面程序设计的基础和精髓。

第六章介绍了Java语言多媒体技术。多媒体技术是Java改变WWW静态文本的关键技术,本章从文本处理、图形处理、图象处理、声音处理以及多媒体声像合成技术应用等方面作了全面介绍。

第七章介绍了Java语言异常处理。解释了异常及异常对象的概念、介绍了异常处理机制,异常类的树结构、异常的产生、捕获与处理、怎样利用Java开发环境定义自己的异常。

第八章介绍了Java多线程程序设计方法。从操作系统原理的角度分析了线程的概念、线程的生命周期、线程组、线程的同步机制。

第九章介绍了Java网络程序设计方法。其中包括:Java网络程序设计概念、数据通信,用

URL 类访问网络资源、句柄以及网络安全措施。

第十章介绍了编写 Java Server 的方法。其中包括:Server Socket 类、回声——一个 Java 服务器的实例、回声服务器的客户端程序。

第十一章介绍了 Java 语言与 HTML 语言的接口。其中包括:HTML 语言概述、HTML 语言的特性、HTML 文件的编辑和浏览、HTML 语言的标签命令;Java applet 与 HTML 文件的连接、Java applet、Java application 和 HTML 的混合编程实例。

第十二章介绍了 Java 语言与 C 语言的接口。其中包括:Java 与 C 的连接、Java 与 C 的连接步骤、Java 与 C 的参数传递和返回值。

本书每一章的最后两部分都是小结和习题。每一章的小结从不同角度概述了该章的主要内容、重点和难点;习题从概念、编程的不同方面帮助读者检查学习效果,巩固已学的知识,进一步提高编程能力。

本书编写过程中,刘传嘉先生从国外及时寄来了我们所需的参考书籍,刘厚嘉女士、武汉大学图书馆副馆长燕今伟、技术部主任王玖、武汉大学图书情报学院实验中心主任沈祥兴等都给予了大力支持和帮助,在此表示最诚挚的谢意。

本书第一章、第八章、第十一章、第十二章由董慧编写;第二章、第四章由张俊编写;第六章、第七章由徐翰伟编写;第五章、第十章由李旭晖编写;第三章、第九章由陆伟编写;全书由董慧统稿。限于水平和时间有限,错误和不妥之处在所难免,敬请广大读者批评和指正。

董 慧

1997 年 5 月于武昌珞珈山

目 录

第一章 Java 的开发环境	(1)
1.1 Java 的运行环境	(1)
1.1.1 Java 技术的现状	(1)
1.1.2 Java 技术的特性	(3)
1.1.3 Java 技术的应用前景	(5)
1.1.4 Java 开发及执行环境	(5)
1.2 Java Developers Kit (JDK)	(7)
1.2.1 WWW	(7)
1.2.2 JDK	(8)
1.2.3 Java applet	(22)
1.2.4 Java application	(23)
1.3 HotJava 浏览器	(24)
1.3.1 HotJava 的功能	(24)
1.3.2 HotJava 的特点	(25)
1.3.2 HotJava 的安全机制	(25)
1.3.4 在 HotJava 页面中创建 applet	(26)
1.4 小结	(27)
1.5 习题	(27)
第二章 数据类型、变量和表达式	(28)
2.1 基本数据类型	(28)
2.1.1 整型	(28)
2.1.2 浮点型	(29)
2.1.3 字符型	(29)
2.1.4 布尔型	(29)
2.1.5 基本类型的包装类	(30)
2.2 数组	(30)
2.2.1 数组的建立与初始化	(30)
2.2.2 数组的访问	(31)
2.3 字符串	(32)
2.3.1 字符串的建立	(32)
2.3.2 字符串的比较	(33)
2.3.3 串操作	(34)
2.4 类型转换	(36)

2.4.1	基本类型之间的转换	(36)
2.4.2	引用类型之间的转换	(37)
2.4.3	字符串与其他类型之间的转换	(38)
2.5	变量	(39)
2.5.1	Java 符号集与变量声明	(39)
2.5.2	作用域	(40)
2.6	运算符和表达式	(42)
2.6.1	算术运算符	(42)
2.6.2	关系运算符	(44)
2.6.3	布尔运算符	(44)
2.6.4	赋值运算符	(45)
2.6.5	位运算符	(45)
2.6.6	其他运算符	(47)
2.6.7	运算符的优先级	(49)
2.6.8	表达式	(49)
2.7	小结	(50)
2.8	习题	(50)
第三章	流程控制	(52)
3.1	Java 与 C 及 C++ 的差别	(52)
3.2	条件语句	(53)
3.2.1	简单条件语句	(53)
3.2.2	复合条件语句	(53)
3.3	循环语句	(54)
3.3.1	for 语句	(54)
3.3.2	while 语句	(55)
3.3.3	do - while 语句	(55)
3.3.4	三种循环语句的比较	(55)
3.4	开关语句	(56)
3.5	转移语句	(57)
3.5.1	break 语句	(57)
3.5.2	continue 语句	(58)
3.5.3	return 语句	(59)
3.6	小结	(60)
3.7	习题	(60)
第四章	Java 的面向对象程序设计	(61)
4.1	概述	(61)
4.2	对象与类	(62)
4.2.1	类说明	(62)

4.2.2	类体	(64)
4.2.3	对象的创建与取消	(72)
4.2.4	对象的使用	(73)
4.3	接口	(74)
4.3.1	定义接口	(75)
4.3.2	使用接口	(75)
4.4	继承	(76)
4.4.1	Java 的继承	(77)
4.4.2	继承的实现	(78)
4.4.3	作为程序设计方法的继承	(81)
4.5	包	(83)
4.5.1	包的定义	(84)
4.5.2	引入包	(84)
4.5.3	Java 的 API 结构	(85)
4.6	对象容器类的利用	(86)
4.6.1	向量	(86)
4.6.2	哈希表	(89)
4.6.3	栈	(92)
4.7	小结	(93)
4.8	习题	(94)
第五章	Java 的用户界面	(95)
5.1	Java 的用户事件处理	(95)
5.2	基本用户界面	(97)
5.2.1	标签	(98)
5.2.2	文本行和文本域	(98)
5.2.3	计算器: 一个用户界面实例	(100)
5.2.4	按钮	(103)
5.2.5	复选框	(104)
5.2.6	列表框	(106)
5.2.7	选择框	(109)
5.2.8	滚动条	(110)
5.2.9	GUI 组件的对等体	(112)
5.3	高级用户界面	(113)
5.3.1	颜色和字体	(113)
5.3.2	面板	(115)
5.3.3	窗口、框架和对话框	(116)
5.3.4	菜单	(117)
5.3.5	新的计算器	(119)
5.3.6	布局管理器	(126)

5.4	小结	(132)
5.5	习题	(133)
第六章	Java 语言中的多媒体技术	(134)
6.1	多媒体技术概述	(134)
6.1.1	多维性	(134)
6.1.2	集成性	(134)
6.1.3	交互性	(135)
6.2	文本处理	(138)
6.2.1	文本的标准输入/输出	(135)
6.2.2	绘制字符串	(137)
6.3	图形处理	(139)
6.3.1	框架图形原语	(139)
6.3.2	填充图形原语	(140)
6.3.3	文本图象绘制方法	(141)
6.3.4	属性控制方法	(142)
6.4	图象处理	(143)
6.4.1	图象的下载与监测技术	(143)
6.4.2	图象的创建及其属性	(146)
6.4.3	异步图象的接口	(147)
6.4.4	图象的低级处理技术	(150)
6.5	声音处理	(157)
6.5.1	声音效果	(157)
6.5.2	声音播放	(158)
6.6	多媒体声像合成技术与应用	(159)
6.6.1	动画制作原理	(159)
6.6.2	闪烁	(160)
6.6.3	声音的应用	(162)
6.6.4	应用实例的简要分析	(162)
6.7	小结	(167)
6.8	习题	(168)
第七章	异常处理	(169)
7.1	异常及异常对象	(169)
7.2	异常处理机制	(170)
7.3	异常类的继承树结构	(173)
7.4	异常的产生、捕获与处理	(176)
7.4.1	异常的抛出:throw 语句	(176)
7.4.2	异常的捕获与处理:try - catch 结构	(179)
7.4.3	finally 代码段	(181)

7.5	定义自己的异常	(182)
7.6	小结	(187)
7.7	习题	(188)
第八章	多线程程序设计	(189)
8.1	线程概念	(189)
8.2	线程的生命周期	(190)
8.2.1	线程状态和线程控制	(190)
8.2.2	线程的创建	(191)
8.2.3	线程的启动和终止	(198)
8.2.4	线程的挂起和恢复	(198)
8.2.5	线程的优先级和线程的调度	(200)
8.3	线程组	(202)
8.3.1	线程组类的管理方法	(203)
8.3.2	组属性操作方法	(204)
8.3.3	涉及组中全部线程的方法	(205)
8.3.4	访问控制	(206)
8.4	线程的同步机制	(206)
8.4.1	临界区和管程	(208)
8.4.2	notify() 和 wait() 方法及其应用	(210)
8.4.3	死锁	(211)
8.5	小结	(212)
8.6	习题	(216)
第九章	Java 网络程序设计	(217)
9.1	Java 的 Socket 通信机制	(217)
9.1.1	何为 Socket	(217)
9.1.2	流	(219)
9.1.3	面向连接的通信方式	(223)
9.1.4	数据流	(224)
9.1.5	管道流	(227)
9.2	数据报通信	(231)
9.2.1	何为数据报	(231)
9.2.2	DatagramPacket 和 DatagramSocket	(232)
9.2.3	发送数据报	(233)
9.2.4	接收数据报	(234)
9.3	用 URL 访问网络资源	(235)
9.3.1	何为 URL	(235)
9.3.2	利用 URL 类访问网络资源	(235)
9.3.3	使用 URLConnection 类访问网络资源	(238)

9.4	句柄	(242)
9.4.1	内容句柄	(242)
9.4.2	协议句柄	(246)
9.5	小结	(248)
9.6	习题	(248)
第十章	编写服务器程序	(250)
10.1	Server Socket 类	(250)
10.2	回声：一个 Java 服务器的实例	(251)
10.2.1	回声设计思路	(251)
10.2.2	回声服务器的实现	(252)
10.2.3	改进回声服务器	(257)
10.3	回声服务器的客户端程序	(258)
10.3.1	EchoClient 的设计与实现	(258)
10.3.2	与非 HTTP 服务器的接口	(260)
10.4	小结	(260)
10.5	习题	(261)
第十一章	Java 与 HTML 的接口	(262)
11.1	HTML 的特性	(262)
11.2	HTML 标签结构	(263)
11.3	HTML 文件的编辑与浏览	(263)
11.3.1	编辑 HTML 文件	(263)
11.3.2	浏览 HTML 文件	(263)
11.4	HTML 的标签命令	(263)
11.4.1	文件结构命令	(263)
11.4.2	区段格式命令	(264)
11.4.3	下锚链结命令	(267)
11.4.4	字符格式命令	(268)
11.4.5	图象命令	(269)
11.4.6	列表命令	(270)
11.4.7	表格命令	(270)
11.4.8	背景和前景颜色变化	(271)
11.5	Java applet 和 HTML 文件的连接	(272)
11.5.1	HTML APPLET 语句	(273)
11.5.2	HTML APPLET 属性设置	(273)
11.6	Java applet 与 HTML 文件参数的传递	(274)
11.6.1	Tumbling Duke	(276)
11.7	Java applet、Java application 和 HTML 混合编程	(280)
11.7.1	Java applet 和 Java application 的差别	(280)

11.7.2	Java applet 和 Java application 混合编程的关键	(281)
11.7.3	Java applet、Java application 和 HTML 混合编程实例	(282)
11.8	小结	(284)
11.9	习题	(284)
第十二章	Java 与 C 的接口	(286)
12.1	Java 与 C 的连接	(286)
12.2	Java 与 C 的连接步骤	(287)
12.3	Java 与 C 的参数传递与返回值	(290)
12.3.1	自动参数	(290)
12.3.2	简单数据类型的参数及返回值	(291)
12.3.3	复杂数据类型的参数及返回值	(291)
12.4	小结	(292)
12.5	习题	(292)
参考文献		(293)

第一章 Java 的开发环境

Java 的出现带来了 WWW 革命,Java 不仅是产品,而且是新的网络操作系统。Java 将会成为 90 年代的超级语言,它将无处不在。Java 的成功意味着计算机成为 Internet 和 WWW 的外设,Java 营造了一个开放环境。

1.1 Java 的运行环境

1.1.1 Java 技术的现状

Java 从起源到发展经历了一个带有浓厚戏剧性色彩的过程,Oak —— 橡树之星,First Person —— 山重水复,www 与 Mosaic —— 它山之石,Internet Play —— 柳暗花明。从 1991 年 3 月,Sun 成立了一个由 Gosling、Naughton 和 Sheridan 三人组成的秘密开发小组,1994 年,Sun 公司创始人之一的 Bill Joy 加盟,1995 年 1 月,Gosling 所完成的 Oak 版本被重新命名为 Java,Naughton 完成的 WebRunner 被命名为 HotJava,Joy 力排众议,在 Internet 上免费发布 Java 语言规范及其开发环境,1995 年 5 月,alpha 版本的 Java 和 HotJava 在 Internet 上正式发布,同年秋,Netscape 获准在其浏览器 2.0 的 32 位版本中支持 Java 语言,随后,除 Sun 和 Netscape 外,共有 28 家著名的计算机公司先后宣布支持 Java,到现在,已是处处都是 Java 声,其发展的势头真有点儿“势不可挡”。Java 已不仅仅是一种语言,而是在逐渐形成一门技术。它的内涵包括:

- Java 软件技术:Java 语言及其编程技术、Java 操作系统(JavaOS)、Java 开发环境与开发工具、Java 数据库接口 JDBC、Java 软件包等等;
- Java 虚拟机及其规范;
- Java 嵌入技术:Java 芯片,基于 Java 技术的新型计算机系统(网络计算机 NC、基于网络计算的 personal 计算机 NetPC、网络计算机体系结构 NCA、网络工作站 NetWork Station,...);
- Java 计算(Java Computing):基于 Java 处理功能的信息综合处理系统,...

这些 Java 技术,其应用范围已不仅仅在信息领域,而是涉及到电器(把 Java 芯片 PicoJava 嵌入移动电话、WebTV 或 InternetTV、洗衣机中等)、信息综合服务、科学计算、软件生产等领域,显示出 Java 技术的广阔的应用前景,因此受到了各类用户的肯定与欢迎。在信息产业界,Java 一推出,就引起了震动。以 IBM 为代表提出的“以网络当中心的计算模式”的“网络计算中心”,认为它将逐渐成为 Internet 上受欢迎的开发与编程语言。一些著名的计算机公司纷纷购买了 Java 语言的使用权,如 IBM、Netscape、Novell、Oracle、Apple、DEC、SGI 等。因此,Java 被美国的一些著名杂志评为 1995 年十大优秀科技产品。1997 年初,美国著名的《Datamation》杂志经过信息系统管理人员的评选,推荐出一批 1996 年计算机市场中最佳新产品,Sun 微系统公司的 Java 经过角逐,最后,击败微软的 Visual Basic 4.0 企业版、Visual C++ 4.0,获得应用开发金牌。随之出现了大量用 Java 编写的软件产品,受到工业界的重视与好评,认为“Java 是八十年代以来计算机界的一件大事”。就连微软总裁比尔·盖茨悄悄地观察了一段时间后,也不无感

慨地说:“Java 是长时间以来最卓越的程序设计语言。”他确定微软整个软件开发的开发战略会从 PC 单机时代向着以网络为中心的计算时代转移。而购买 Java 的使用权,则是他的重大战略决策的实施。Apple、HP、IBM、Microsoft、Novell、SGI、Tandem 等公司均计划将 Java 并入各自开发的操作系统中。而负责开发并推广 Java 技术的 Sunsoft 公司,将用颁发许可证的办法来允许各家公司把 Java 虚拟机和 Java 的 Applets 类库嵌入到他们的操作系统中。这样,各类开发人员就能更容易地选择多种平台 Java 语言来编程,不同的用户也就可以脱离 Web 浏览器来运行 Java 应用程序。这样,无疑,Java 语言的应用开拓了极为广阔的前景。有人曾预言:Java 将是网络上的“世界语”,今后所有的用其他语言编写的软件系统都将用 Java 语言来改写,它会永远改变人们编写软件的方式。Internet 将改变我们的世界观,而 Java 将改变我们的方法论。因此,Java 的诞生必将对整个计算机产业发生深远的影响,对传统的计算模型提出新的挑战,“Java 为 Internet 和 WWW 开辟了一个崭新的时代。”

在学术界和教育界,教授们用它来传播学科发展动态、先进的学术观念、作为获取新知识的工具,甚至在 Internet 上用它来进行继续教育、学术讨论等重要活动。近来,在美国大学和各大院校中,Java 语言获得相当的声誉,对 Internet 和 Web 而言,Java 已成为简单方便和合适的主流开发语言。人们普遍认为:

(1)Java 将成为流行的主流语言。由于网络技术和 Web 的飞速发展,在高等教育中,Java 语言已扮演了重要角色。在近期内,共同的要求是必须开设 Java 语言课程,以普及和推广其应用;今后,作为一种主流程序设计语言,必须培训学生具有能用它来进行开发、科学研究的能力,更好地激励 Web 的发展。纽约州大学 1995 年 9 月在计算机科学会上把 Java 语言当作主修课程介绍。Java 语言奇迹般地充满了所有的文化领域,它是计算机程序设计和计算机科学及计算机设计的发展方向。

(2)Java 比 C++ 更好。计算机科学著名人士 David Dobkin 说:“用哪一种程序设计语言才能够使我们给学生讲清楚计算机科学的基本概念呢?”,他认为:“Java 几乎是十全十美没有缺陷的,并且,我们现在坚信:Java 可能将成为世界上最好的程序设计语言。”San Diego 州立大学数学科学系教授 Roger Whitney 先用 C 语言和 C++ 语言教他的“客户服务计算课程”,然后再用 Java 语言教,教学效果是成功和惊人的,“Java 语言是一门高水平的语言”。Whitney 说:“它没有指针运算,仅仅有引用,并且 Java 有一个标准类库,学生们可以在任何平台运用它。他们用 C 和 C++ 语言,要花费很长时间和精力才能掌握指针。”

(3)用 Java 作为科学研究工作的必备工具。Java 语言通过自动垃圾收集管理,大大简化了内存管理工作,使用方便。它的许多优势和特征使它成为开发者乐意接收的语言,而且逐渐成为科学研究工作的必备工具。

对软件开发技术人员来说,Java 的出现,将大大解除软件编程人员的“搬家改码”的烦恼,Java 能“一次编程,到处运行”,颇受软件开发人员的青睐。Java 语言之所以对未来的软件开发产生影响,其原因主要有如下几方面:

(1)软件的需求分析:在 MIS 系统中,用户的需求动态性强,目前,许多程序设计语言,很难适应动态需求的变化,这也是导致许多 MIS 系统很难走向实用的原因之一。Java 一改传统程序设计的风格,它可将用户的需求进行动态的、可视化的描述,用户的各种各样的需求,不受地区、行业、部门、爱好、时间的影响,都可描述清楚;

(2)软件的开发方法:Java 语言是一种纯面向目标的开发工具,完全可以用 OO 的技术与方法来进行面向对象的程序设计;

(3)Java 语言动画效果远比 GUI 技术更加逼真,尤其是利用 WWW 提供的巨大动画资源空间,共享全世界的动态画面资源;

(4)软件最终产品:用 Java 语言开发的软件可以具有可视化、可听化、可操作化的效果,它比电视、电影的效果更为理想,它能真正做到“即时、交互、动画与动作”,要停即停,要动即动,反复选择,别具一格,这是电影与电视播放过程中难以实现的。

在金融界,Java 已成为华尔街相互沟通的“共同语”,他们纷纷推出用 Java 编写的各种金融软件(如股票信息查询系统、金融货币支持清算系统、电子钱包等)。一项技术的出现,能否为人们所接受,能否占领市场,产生的经济效益的高低是检验优劣的一项重要标准,金融界的看法,有时则起着决定性的作用。华尔街对 Java 的态度说明这种科技是值得推广的。

在工业界,对 Java 的热衷可以从 1996 年 5 月 29~31 日在美国旧金山召开的全世界 Java 语言开发大会——Java One Developer Conference 得到验证,出席会议的代表多达 6500 多人,来自工业界的超过一半。有人评价说:“这是近年来计算机界最光辉的一次盛会。”一些工业界的老总们相信看好 Java 语言,认为它的使用将会引起一场软件革命。从软件的设计风格、设计方法、设计目标到设计过程,都会产生彻底的变革,“甚至会改变此星球的生活方式”。Dta 咨询公司的高级软件工程师 Rich Kadel 说:“Java 不仅仅是一种程序设计语言,更是现代化软件再实现的基础;Java 还是未来的新型 OS 的核心;将会出现 Java 芯片;将构成各种应用软件的开发平台与实现环境,是人们必不可少的开发工具,…”“现在第三方的开发商都团结在 Java 大旗周围了!”

1.1.2 Java 技术的特性

Java 为什么能让上述各界人士达到共识?为什么如同旋风一时吹遍全球?为什么能称得上是一次技术革命?归根结底,还是由于 Java 历经磨难,总结了计算机发展中的经验与教训,形成了自身的许多突出特点所决定的。它有哪些特点?归纳起来主要有如下几方面。

1. 简单性

从 Java 语言起源与发展来看,一方面,它由 C++ 衍生而来,其语言风格与 C++ 十分相似;另一方面,Java 又比 C++ 简单,它抛弃了 C++ 中一些不是绝对必要的功能,如头文件、预处理器、指针、结构、联合、隐式的类型转换、操作符重载等,这样就使 Java 比 C++ 更容易学习,其程序的可读性更强。

2. 纯面向对象

Java 是一种完全面向对象的程序设计语言,它除了数值、布尔和字符三个基本数据类型外的其它类型都是对象,Java 程序代码以类的形式组成,由类定义对象的各种状态和行为,Java 抛弃了 C++ 中的非面向对象特性,如结构和函数调用,Java 也不再支持全局变量。Java 支持继承特性,但它只支持类的单重继承,即每个类只能从一个类中继承。Java 支持界面,界面允许程序员定义方法但又不主动实现,一个类可以实现多个界面,利用界面可以得到多重继承的许多优点而又没有多重继承的问题。

3. 可移植性

Java 定义出自己的一套虚拟机,以及这套虚拟机上所使用的机器码——JavaBytecode。关于虚拟机我们留在后面讨论。这里主要介绍 Bytecode 的工作过程来说明与机器无关。Bytecode 十分类似于机器指令,但它不是为某个特定机器定义的,因此一般不能直接在某个平台上执行,而需要由 Java 运行系统中的解释器来解释执行,即首先由编译器将 Java 程序编译成字

节码,然后再由 Java 运行系统解释执行 Bytecode。由于 Java 通过预先将源代码编译为接近于机器指令的字节码,有效地克服了传统解释型语言的性能瓶颈,又由于解释执行 Bytecode 有需要 Java 运行系统不需要具体某个平台的支持,因此,保证了运行环境与某个具体机器的无关性。

4. 稳定性与安全性

分布或计算环境要求软件具有高度的稳定性与安全性。为保证稳定性,Java 采取了三项措施,首先,Java 不支持指针数据类型,这样,程序员便不再能够凭借指针在任意内存空间,甚至是 OS 自由存取空间中“漫游”;其次,Java 提供了数组下标的检查机制,从而使网络“黑客”们无法构造出类似 C 和 C++ 语言所支持的指针;最后,Java 还提供了内存管理机制,即一个自动的“内存垃圾”搜集程序。为了安全,Java 除打算在 Bytecode 的传输过程中使用公开密钥加密机制 (PKC) 外,还提供了四级安全性保障机制,字节码校验器 (Bytecode Verifier)、类装载器 (Class loader)、运行时内存布局、文件访问机制等。

5. 结构中立性

Java 语言的设计不针对某种具体结构,采取了结构中立。除了上面提到的 Bytecode 外,还制定了完全统一的语言文本,如 Java 的基本数据类型不会随目标机的变化而变化,一个整型总是 32 位,一个长整型总是 64 位,不像 C 和 C++ 那样,不同的目标机,整型量的长度是不同的。同时,为了使 Java 的应用程序能不依赖于具体系统,Java 语言环境还提供了用于访问底层 OS 功能的扩展类库。

6. 高性能

一般情况下,可移植性、稳定性和安全性几乎总是以牺牲性能为代价的,解释型语言的执行效率一般也要低于直接执行源码的速度。但是,为了弥补这些性能受到的影响 Java 采取了一系列有效措施。这些措施包括:

- 多线程。线程是现代 OS 所提出的一个新概念。线程,是一种轻进程,比传统进程小的一种可并发执行的执行单位。线程的概念提高了程序执行的并发度,从而提高系统效率。Java 的多线程机制主要体现在两个方面:首先,Java 环境本身是多线程的,它可以利用系统的空闲时间来执行诸如必要的垃圾清除和一般性的系统维护等操作;第二,Java 还提供了对多线程的语言级支持,即编程人员可以利用多线程编程接口编写支持多线程的应用程序,从而提高程序的执行效率。
- 高效的字节码。正如前面介绍的,字节码为 Java 与机器无关提供了舞台,同时,由于 Java 的字节码格式的设计充分考虑了性能因素,其格式非常简单,这使得经由 Java 解释器执行时可产生高效的机器码。
- 及时编译(可选)。在运行时把字节码编译成机器码,在一开始会有一个编译字节码的延迟过程,而这种代码仍然是可移植的。
- 嵌入 C 代码(可选)。在 Java 中可以直接嵌入 C 代码,嵌入 C 后的运行速度是最理想的,但给编程人员带来了额外负担,同时也会降低代码的可移植性。

7. 动态特性

Java 的动态特性是其面向对象设计的延伸。Java 的动态特性使得 Java 程序能够适应不断变化的执行环境。这种适应性来自于 Java 程序的构造,Java 程序的基本组成单位是类,而 Java 分类又是运行时动态装载的,这使得 Java 可以在分布环境中动态地维护应用程序及其支持类库之间的一致性。这样,对于 Java 而言,其支持类库升级之后,相应的应用程序不必重新编

译,也一样可以利用升级后类库的新增功能。Java 的动态性还体现在其对动态性使你能够真正拥有“即插即用”(plug-and-play)的软件模块功能,这就是 Java 动态特性的价值所在。

8. 分布性

分布性主要是指数据分布和操作分布。数据分布是指数据可以分散存放于网络上的不同主机;操作分布则把计算分散由不同主机进行处理。Java 支持 WWW 客户机/服务器计算模式。通过 Java 提供的类库可以处理 TCP/IP 协议,用户可以通过 URL 地址在网络上访问对象,其访问方式如同访问本地文件系统。

以上这些特性可以看出,Java 确是一门适合 Internet 和分布式环境的技术。

1.1.3 Java 技术的应用前景

由于 Java 技术本身所具有的特性,以及各界人士对 Java 技术的认可,其应用前景极为宽广。大体上可以从以下几个方面考虑其应用:

- 所有面向对象的应用开发,包括面向对象的事件描述、处理、综合、分析等;
- 计算过程的可视化、可听化、可操作化的软件的开发;
- 动态画面的设计,包括图象图形的调用;
- 交互操作的操作(选择交互、定向交互、控制流程等);
- Internet 的系统管理功能模块的设计,包括 Web 页面的动态设计、多媒体设计、管理和交互操作设计等;
- Intranet 上的软件开发(直接面向企业内部用户的软件开发);
- 与各类数据库连接查询的 SQL 语句的实现;
- 家用电器的嵌入技术(基于 Java 芯片的技术,尤其是 WebTV 或 InternetTV 的开发使用);
- 其他应用类型的程序。

1.1.4 Java 开发及执行环境

许多程序设计语言通过对源程序进行编译、链接和装入,直接生成可执行的代码。而 Java 的目标代码是字节码,一方面,字节码不是直接针对某个具体的平台,在执行之前,需要将字节码转换为本机代码;另一方面,为了实现其动态性和安全性,Java 编译器没有将变量和方法的引用直接编译为具体的内存引用,也没有确定程序执行过程中的内存布局,而是将符号引用信息保留在字节码中,这样类的装载程序运行时需要类、安排程序运行时的内存布局、确定方法调用所需要的实际地址,最后也是由运行系统来控制执行代码。可见,无论是 Java 运用还是 applet,在执行时都需要 Java 运行系统的支持,对于 Java 的运用,一般都需要 Java 解释器;而对于 Java 的 applet,一般都需要与 Java 兼容的 Web 浏览器,即 Java 兼容浏览器。

Java 运行系统一般由类装配器、字节码验证器、解释器、代码生成器和运行支持库组成。

其组成如图 1.1 所示。

Java 运行系统执行字节码的过程可分三步:代码的装入、代码的验证和代码的执行。代码的装入工作由类装配器完成,类装配器负责装入程序运行时需要的所有代码,其中包括程序代码中调用到的所有类。当类装配器装入一个类时,该类被放在自己的名字空间,除了通过符号而引用其它类外,该类不能影响其他类所在的空间。本地计算机上的所有类都被安排在同一地址空间,而所有从外部引进的类都有一个自己独立的名称空间,这种作法,即保证了本地类