

实用 Java 教程



主编 王能

上海交通大学出版社

内容简介

本书全面介绍了 Java 语言的基本概况、特点、数据类型及基本语句,系统地讲解了 Java 语言在 Internet 上的应用、多媒体编程及交互界面的设计,特别是系统地介绍了面向对象的程序设计方法以及 Java 语言的高级编程技术。本书理论联系实际、深入浅出、通俗易懂。

本书在介绍基本概念的同时,列举了大量典型而有意义的程序实例。读者通过阅读这些实例并结合上机操作,能较快地掌握 Java 语言的程序设计,并了解面向对象的设计方法。

本书可作为高等院校计算机专业或其他专业程序设计的教材或教学参考书,也可作为计算机应用科技人员的自学或培训教材。

实用 Java 教程

上海交通大学出版社出版、发行

上海市番禺路 877 号 邮政编码 200030

全国新华书店经销

昆山亭林印刷总厂·印刷

开本:787×1092(毫米)1/16 印张:13.75 字数:335 000

版次:1997 年 5 月 第 1 版 印次:1997 年 6 月 第 1 次

印数:1—4 000

ISBN 7-313-01857-6/TP·339 定价:22.00 元

前 言

Internet 是目前全球范围最大、应用最广的计算机网络，它的应用开发已成为当前计算机技术领域的热点。Sun 公司开发的面向对象的编程语言 Java 以其简单性、分布性、安全可靠、可移植性、平台独立性以及对多线程的支持等优势而流行全球，成为最受欢迎的 Internet 编程语言。本书是作为高等学校计算机专业本科生、大专生的 Java 语言教材进行编写的，但对学习 Java 编程的各类人士都是十分有价值的。

在本书中，根据学习计算机语言的特点和规律，首先简要介绍了 Java 语言的产生背景、特点、面向对象的基本概念及编程技术，然后由浅入深详细介绍了 Java 语言的基本语法、Java 中的面向对象编程技术、Java 应用程序(Application)及被广泛用于 Internet 信息发布的 Java 小程序(Applet)、对多媒体支持、交互界面设计等。在此基础上，介绍了 Java 语言中的包(Packages)、接口(Interfaces)、流(Stream)以及 Java 中的多线程机制、异常处理和原生方法(native method)等深入编程技术。

本书每章的后面都备有练习或思考题供学生做实际练习。由于 Java 是一种较为年轻的编程语言，在国内一些术语还未尽统一，在本书附录中给出了 Java 中的关键词汇和术语解释以及 Java 中的类库，相信对读者全面掌握 Java 语言是有益的。

作为一门编程语言的教材，为了方便读者理解 Java 语言的有关概念和掌握其程序设计技巧，书中常以实例加以介绍说明，这也是本书的特点之一，对有一定编程基础的读者会更受欢迎。

本教材由华东师范大学王能教授主编，中国上海测试中心洪孝诒编写了第 1 章，苏州职工大学冯矢勇编写了第 2 章，华东师范大学的张卫编写了第 3 章，顾振宇编写了第 4 章，刘啸林编写了第 5 章，张进兵编写了第 6 章，沈富可编写了第 7 章，洪孝诒及张进兵编写了附录，王能教授审定了全书初稿。

由于水平、时间所限，加以 Java 本身也在不断发展，不妥之处在所难免，欢迎读者批评指正。

编者

1997 年 1 月

目 录

第1章 引 论.....	1
1.1 Java 语言概述.....	1
1.1.1 Java 语言产生的背景和现状.....	1
1.1.2 Java 语言的前景.....	4
1.1.3 Java 语言的开发及其执行环境.....	5
1.2 面向对象编程及 Java 语言的特点.....	9
练习题.....	12
第2章 Java 的基本语法.....	13
2.1 词法与句法.....	13
2.1.1 单词符号.....	13
2.1.2 空白符.....	13
2.1.3 注释.....	14
2.1.4 关键字.....	14
2.1.5 标识符.....	15
2.1.6 常量.....	15
2.1.7 分隔符.....	18
2.1.8 操作符.....	18
2.1.9 一个实例程序.....	18
2.1.10 与 C/C++ 的差异.....	19
2.2 数据类型与变量.....	19
2.2.1 简单类型.....	20
2.2.2 变量的声明、定义和使用.....	21
2.2.3 数组.....	22
2.2.4 数据类型的转换.....	24
2.2.5 变量的作用域.....	25
2.2.6 简单类型与对象的关系.....	25
2.2.7 与 C/C++ 的差异.....	26
2.3 操作符与表达式.....	26
2.3.1 算术操作符.....	26
2.3.2 赋值操作符与复合赋值操作符.....	28
2.3.3 按位和移位操作符.....	29
2.3.4 关系操作符.....	30
2.3.5 逻辑操作符.....	31
2.3.6 其他操作符.....	31

2.3.7 操作符的优先级和结合规则.....	32
2.3.8 表达式和表达式语句.....	33
2.3.9 与 C/C++ 的差异.....	33
2.4 控制流程.....	34
2.4.1 语句的分类.....	34
2.4.2 if...else 条件语句.....	34
2.4.3 switch 开关语句.....	35
2.4.4 while 和 do...while 循环语句.....	37
2.4.5 for 循环语句.....	38
2.4.6 转移语句.....	38
2.4.7 与 C/C++ 的差异.....	39
练习题.....	39
第 3 章 面向对象编程	40
3.1 类和对象.....	40
3.1.1 创建对象.....	40
3.1.2 访问、设置类和实例变量.....	41
3.1.3 方法的调用.....	42
3.1.4 引用对象.....	43
3.1.5 对象类型和基本类型的强制转换.....	43
3.1.6 对象的比较操作.....	44
3.2 类和类的方法.....	45
3.2.1 类的定义.....	46
3.2.2 实例和类变量的创建.....	46
3.2.3 方法的定义.....	48
3.2.4 编写 Java 应用程序.....	50
3.2.5 Java 应用程序和命令行的参数.....	51
3.2.6 方法重载.....	52
3.2.7 构造方法.....	53
3.2.8 方法覆盖.....	56
3.2.9 析构方法.....	59
3.2.10 Java 类库.....	59
3.3 编写 Applet.....	59
3.3.1 Applet 和应用程序的区别.....	60
3.3.2 创建 Applet.....	60
3.3.3 HTML 的 Applet 标记.....	63
3.3.4 向 Applet 传递参数.....	64
练习题.....	65
第 4 章 Java 对多媒体的支持	66
4.1 图形与文本.....	66

4.1.1 图形的绘制.....	66
4.1.2 文本与字体.....	72
4.1.3 颜色的设置.....	74
4.2 图像与声音.....	76
4.2.1 图像文件的显示.....	76
4.2.2 声音文件的播放.....	79
4.3 动画制作.....	81
4.3.1 一个简单实例.....	81
4.3.2 引入线程机制.....	82
4.3.3 初识闪烁问题.....	85
4.3.4 放映图像.....	88
4.3.5 使用媒体跟踪器.....	92
4.3.6 移动型动画.....	94
4.3.7 双缓冲技术.....	97
练习题.....	100
第5章 输入、界面和网络.....	102
5.1 接收鼠标、键盘的输入.....	102
5.1.1 鼠标的按键.....	102
5.1.2 鼠标的移动.....	104
5.1.3 键盘的输入.....	105
5.1.4 事件处理器(Event Handler).....	106
5.2 用户界面设计.....	107
5.2.1 抽象窗口工具箱(AWT)概述.....	107
5.2.2 UI 基本元素.....	108
5.2.3 版面设置.....	116
5.2.4 事件的处理.....	122
5.2.5 窗口构造构件.....	124
5.3 网络编程.....	130
5.3.1 网络资源的使用.....	130
5.3.2 Socket 编程.....	133
5.3.3 数据报.....	135
练习题.....	136
第6章 深入编程 (一).....	137
6.1 访问控制.....	137
6.1.1 方法和变量的存取控制.....	137
6.1.2 存取实例变量.....	140
6.1.3 类变量和类方法.....	141
6.1.4 final 修饰符.....	142
6.1.5 抽象类和抽象方法.....	143

6.2 包和接口	144
6.2.1 包	144
6.2.2 接口	149
6.3 流	154
6.3.1 输入流	154
6.3.2 输出流	164
6.3.3 相关的类	170
练习题	171
第7章 深入编程 (二)	172
7.1 异常处理	172
7.1.1 传统编程语言中的异常处理模式	172
7.1.2 Java 中异常定义	172
7.1.3 Java 的异常处理	173
7.1.4 异常的产生	174
7.1.5 方法覆盖中的异常处理	176
7.1.6 finally 子句	177
7.2 多线程机制	178
7.2.1 多线程	178
7.2.2 多线程的优势	179
7.2.3 原子操作	179
7.2.4 原子操作的使用	180
7.2.5 类变量的保护	183
7.2.6 创建和使用线程	184
7.2.7 可执行的接口 Runnable (Runnable Interface)	184
7.2.8 判断线程何时结束	186
7.2.9 线程调度	187
7.3 原生方法和原生库	190
7.3.1 使用的场合	190
7.3.2 使用原生方法的缺点	191
7.3.3 Java 程序优化	191
7.3.4 原生方法	192
练习题	197
附录一 Java 的基本语法	198
附录二 Java 的基本类库	203
java.lang	203
java.util	204
java.io	204
java.net	205
java.awt	205

java.awt.image.....	207
java.awt.peer.....	207
java.applet.....	207

第 1 章 引 论

一个全球性信息网络——Internet 以迅雷之势延伸到世界各地，它正深刻地影响着世界经济、文化、科技等事业的进步和发展。而今网络世界里最具活力的事件，就是在 Internet 上最有影响的 Java 语言的诞生。这个被誉为“最卓越的程序设计语言”，正在对整个信息世界产生深远的影响。许多有识之士都把它视为信息领域一场新的技术革命。为了让读者了解这一点，我们在这一章介绍 Java 语言的产生、影响、开发环境及其特点。从而为以后诸章学习作个先导。

1.1 Java 语言概述

1.1.1 Java 语言产生的背景和现状

1. Internet 网络

Internet 网络是当今最大的国际互联网络，它连接着大量相互独立的计算机和网络，其中包括美国、加拿大、欧洲等一些地区性网络和商业性的国际网络，从而成为“网间网”。它的原型是美国国防部下属的高级研究计划局 ARPA 在 60 年代末所开发的 ARPANET 网络。它采用分布式结构，由各独立的节点组成，信息被分组并加以编号，各分组传输路径可不同，所以即使局部节点或线路出现故障，信息也能正确地传送到目的地。这种处理方式正是军事上所需要的。然而，在另一方面这正好为通信发展与共享资源的实现带来前所未有的好处，随着计算机通信事业的发展和用户对资源共享的需求，使 ARPANET 的技术和规模不断开拓。最初 ARPANET 的项目名为“The Internetworking Project”。所以，以它为基础的世界性网络就称为 Internet 网络。

Internet 所遵循的主要网络协议是 TCP/IP (Transport Control Protocol/Internet Protocol)。TCP/IP 是一套开放性的协议，以至使 Internet 发展成包罗万象的信息海洋，当今它以图、声、文并茂的方式提供各种信息服务。

据统计，到 1996 年 1 月 Internet 连网主机超过 947 万台，用户数目超过 7000 万，并以每月 15% 速度增长。

2. World Wide Web(WWW)

WWW 又称为环球信息网，万维网，Web 等，它最早是由欧洲粒子物理实验室（CERN）的 Tim Berners-Lee 于 1989 年在 Internet 上开发的，目的是为世界各地的物理学家的协作、交换彼此想法提供信息服务，称为 WWW 高级浏览服务。环球信息网包括 WWW 客户机（一个能够阅读超文本文件程序的浏览器）和 WWW 服务器（通常是基于 UNIX 的主机）两个方面，以客户机/服务器(client/server)模式工作。它基于 HTTP(超文本传输协议)的工作过程是这样的：WWW 浏览器通过 TCP/IP 网络对 WWW 服务器提出客户对超媒体(包括文本、图像、声音等)信息访问请求。WWW 服务器产生应答数据、回送应答数据并关闭链接，完成一次基于

HTTP 协议(超文本传输协议)的会话。WWW 浏览器对应答数据进行解释并显示给用户。

1993 年初, 美国伊利诺斯大学 NCSA (国家超级计算机应用中心) 的 Marc Andreessen 在 UNIX 上推出 NCSA Mosaic。作为最初版本的浏览器, 由于它把丰富的彩色和图形直接带到用户面前, 并通过超链(Hyperlink)连接到相应的信息页, 所以很快就流行起来。Mosaic 的出现极大地推动了 Internet 服务业务。后来又相继出现了 Netscape 的 Navigator、Sun 的 HotJava、Microsoft 的 Explorer 等等至少有 20 多种基于各种操作系统的浏览器软件。用户沿着一条称为链的索引浏览网上的数据。

一个 Web 文档可含链接项, 由这些链接项来指定存储在另一处计算机上的 Web 文档。当用户链接到服务器时, 所看到的第一个信息界面就是 WWW 的主页(Home page), 通过 Home page 上定义的超链可访问相关的 WWW 信息。Web 通过如下相关的协议和标准进行信息定位和存取: URL(统一资源定位器); HTTP(超文本传输协议); HTML(超文本标识语言); CGI(公共网关接口)。有关的内容, 将在下面简述。必须指出, HotJava 是一种新型浏览器, 它可以动态装载并运行 Java 语言的可执行模块, 可以在不使用公共网关接口 CGI 的情况下给 Web 增加动态和交互特性。

现在任何团体和个人都可以通过创建 HTML 文档和公布 URL 地址来发布信息。该文档可供任何浏览器浏览。而文档的访问权限由配置文件或防火墙(Firewall)控制。

3. HTML 超文本标识语言

HTML 是遵循 ISO 8879:1986 SGML(标准广义标识语言)的一种标识语言, 它可以用以生成超文本档案。Web 页面利用 HTML 来组织文档和链接文档。HTML 语言可以控制文档的字型、字体大小、空白等, 可嵌入其他的实体如正文、图形、图像、声音等, 也可通过超链引用其他 WWW 网点资源。

HTML 定义了文档(包括文本、图像、动画和声音等)的结构、标识及属性。也可定义文档结构、字体字形、版面设计、链接等。

HTML 的语法结构表示为:

<HTML 标记> 或 <HTML 标记>文本<HTML 标记>

一些 HTML 标记是单标记, 一些 HTML 标记是双标记需成对使用。

HTML 的链接可包括普通文档链接、图形化文档链接及内部文档链接等。链接文档类型包括正文、图形、声音等。

下面简要介绍一些 HTML 标记及其功能:

标记	功能
<HTML>...</HTML>	标记整个 HTML 文件, 即这对标记之间是整个 HTML 文件
<HEAD>...</HEAD>	标记文件的头部信息, 即这对标记之间是文件的总标题等信息
<BODY>...</BODY>	标记正文的内容, 即这对标记之间是 HTML 文件的正文
<HR>	产生一条分隔线
 	在当前位置上换行

<H1>...</H1>	这对标记之间的内容是第一层标题，字体较大
<H2>...</H2>	这对标记之间的内容是第二层标题，字体比上一级标题小
<H3>...</H3>	这对标记之间的内容是第三层标题，字体比上一级标题小
<H4>...</H4>	这对标记之间的内容是第四层标题，字体比上一级标题小
<H5>...</H5>	这对标记之间的内容是第五层标题，字体比上一级标题小
<H6>...</H6>	这对标记之间的内容是第六层标题，字体比上一级标题小
<I>...</I>	斜体字，即这对标记之间的内容显示为斜体
...	粗体字，即这对标记之间的内容显示为粗体
<U>...</U>	文字下面划线
<P>	段落分界标记
...	链接 URL
...	链接电子邮件服务
...	链接另一个文件
...	显示文件名为 xxx.GIF 的图片

4. HTTP(Hyper Text Transfer Protocol)超文本传输协议

超文本传输协议 HTTP 是基于 TCP/IP 之上的协议，是 Web 浏览器和 Web 服务器之间的应用层通信协议。它是 90 年代初出现的一种 Internet 协议。

5. Java 语言的产生及对 Internet 的影响

Java 语言的产生有着曲折离奇的过程，1991 年 Sun 公司为了开拓面包烤箱和交互式电视等一类家用电器市场，成立了以 Gosling，Naughton 等组成的三人小组（家电软件开发小组）从事家电软件开发，并要求该软件产品简洁、安全，而且整个系统必须与平台无关。于是，他们先从改写 C++ 编译器着手。但是结果却无济于事。Gosling 决定自行开发一种新的语言，使这个系统能工作在分布的、异质的网络环境中，完成电子设备间的通信。该语言在编译后产生字节码，然后运行在一个解释器上，只要操作系统配有相应的解释器，就可以与平台无关了。这种新语言起初命名为 Oak(橡树)，后来发现 Oak 已被 Sun 公司另一语言注册，便改名为 Java(太平洋上盛产咖啡的爪哇岛岛名)。

Oak 产品在技术上是成功的，由于经营策略上的问题，结果未能占有家电市场。再说当时伊利诺斯大学的 NCSA 推出的 Mosaic 1.0 版本已用于 WWW，虽是声、图、文并茂，然而却是静态的。为了同时具备动态性，需要开发一种语言。该语言要简炼，运行环境(包括编译)要小，有较好的安全性。Oak 正好能满足这些要求。当时，Sun 公司的创始人 Bill Joy 指定 Gosling 继续完善 Oak 语言，同时 Gosling 意识到 WWW 必须有一个独立于任何硬件、软件平台的浏览器，并且是一个实时性强、安全可靠、有交互功能的浏览器。此项工作由 Naughton 负责，并且定名为 WebRunner。1994 年底，他们各自完成了自己的工作。1995 年 1 月 Oak 改名为 Java。随之 WebRunner 改名为 HotJava。它是用 Java 语言编写的浏览器。1995 年 5

月作为 Alpha 版本的 Java 以及 HotJava 在 Internet 上正式发布。由于它的功能卓越，立即就在产业界引起巨大轰动。Java 地位也进一步确立。

1995 年春，Sun，SGI，Macromedia 三家公司联合制定基于 Java 的应用编程接口 (API)；1995 年秋 Netscape 公司第一个获准在其浏览器 2.0 版本中支持 Java 语言；同年一些著名的计算机公司纷纷支持并购买 Java 语言许可证，包括 IBM，Oracle，Adobe，DEC，SGI，Apple，HP 等公司，Microsoft 虽然不情愿，但还是在他的 Web 浏览器 Explorer 3.0 版增加了对 Java 的支持。不少公司已决定将 Java 放进该公司开发的操作系统，使用户可以不通过 Web 浏览器来运行 Java 应用程序。

另一方面，HotJava 增加了可执行内容，突破了传统浏览器的那种静态模式。允许应用程序在网间传输执行，用户在浏览这些程序时可与网络交互，使 Internet 更具活力。用户不仅能浏览信息，而且能设计交互式的科学实验模拟程序、交互式广告系统、购物系统，可在广泛范围内对各种课题进行交互讨论研究。HotJava 的这一动态特征完全依赖于编程语言 Java。因此，在最近的关于 NC(网络计算机)的技术规范讨论中，Oracle，IBM，SUN 等已将支持 Java 语言作为 NC 的一个基本条件。

经过一年多的使用和改进，Java 1.0 版本于 1996 年初发表。很快为产业界所接受，在推广的过程中，大家公认 Java 无愧为信息界的一颗晶莹的明珠。1996 年 5 月，在美国旧金山召开的世界 Java 语言开发者大会时，出席会议人数达 6000 多人，被称为计算机界光辉的盛会。预示着 Java 将成为计算机事业的下一发展浪潮。

1.1.2 Java 语言的前景

毋庸置疑，Java 语言的产生是软件工业的一场革命，它革新了设计方法、设计风格和设计过程。过去由于互不兼容性，使用户产生不少麻烦和不便。有了 Java 以后得到了极大的改善。甚至在一般情况下，用户只需一个操作系统和一个浏览器就可完成一系列应用和开发。

由于技术的独立性和强大的交互性，Java 软件工业带来了许多根本性的改革，包括微软公司在内，许多有实力的大公司都把 Java 技术作为今后开发新产品的的基础。Java 语言对未来的影响，大致有如下几个方面：

为开放系统在各个计算机领域的应用和发展开辟了道路。以前各公司、厂商的产品由于互不兼容性，各执己见，使各种产品很难达到完善的开放功能。Java 的平台独立性以及操作系统、数据库、网络开发和图像技术等诸方面对 Java 的重视，将大大促进开放系统的发展。

Java 语言是至今最纯、最优秀的面向对象的语言。Java 语言不仅可缩短软件开发周期，并在很大程度上提高了软件质量。同时 Java 在可视化、动态画面设计、交互操作设计及各种功能模块设计方面都会带来全新的局面。

Java 语言在各种管理系统和企业内部网络上的软件开发方面，给人以一种新的景象。难怪华尔街的金融界也投入不少资金来开发相应贸易金融软件。

Java 语言为分布式对象技术提供了良好的环境。Java 的应用程序可以相互协作，可作为一个程序模块使用，就像一个软件的组件(软件 IC)，联合完成软件开发工作。由于可移植性和平台的独立性，使它更优于其他的类似的软件。

最后，必须提及的是 Java 对人们广泛感兴趣的网络计算机的开发和发展提供了强大的支持。

1.1.3 Java 语言的开发及其执行环境

1. 一般讨论

用 Java 语言编写的程序,按其实现的环境主要可分为两类,一类是独立执行的应用程序(Application),另一类是只能在 Java 浏览器里执行的小应用程序(又称 Applet),其实 Applet 不一定是小程序。称它为小应用程序只因为它运行在 web 浏览器上。二者只要作一些小修改就可以实现相互转换了。

(1) 独立执行的程序是以 .Java 为后缀的源程序,它可利用各种文本编辑器编辑,文件名格式为 filename.java。首先对它进行编译(例如,在 SUN 的 JDK 中用命令 javac 进行编译),即

```
javac filename.java
```

编译后生成 filename.class 文件。这个文件可直接用 Java 解释器来解释执行(例如,在 SUN 的 JDK 中用命令 java 进行解释执行),即

```
java filename
```

独立应用程序的执行需要有一个 main()方法集中控制(main 必须被定义为公共类型而且必须是静态的,这与其他 main 方法不一样)。通常返回类型为空(void),即不作为需要返回值的函数,仅作为过程。main()方法负责调用其他函数或过程。

(2) 小应用程序(Applet)通常嵌入到 HTML 的文件中,它使 Web 页面增加交互性和动态性。Applet 不需要 main()方法。

下面是两个简单的程序,分别属于 Application 和 Applet,他们具有相同的功能:在屏幕上打印"Hello World!"

独立执行应用程序 Hello WorldA.java

```
class HelloWorldA {  
    public static void main(String args[]){  
        System.out.println("Hello World!");  
    }  
}
```

这个程序有两个主要方面:

程序的整体被封装在一个类(class)定义中。这里的类名被称为 HelloWorldA。

程序体(这里只有一行)包含在被称为 main()的方法里,与 C 及 C++一样,在程序执行时,它是首选的。

javac 要求每个源文件只有一个公共类,而且公共类的名字和程序文件名(不考虑扩展名.java)应是相同的。

小应用程序 Applet 的例子: HelloWorldB.java

```
import java.applet.*;  
import java.awt.*;  
public class HelloWorldB extends Applet{  
    public void paint(Graphics g) {  
        g.drawString("Hello World!", 20, 50);  
    }  
}
```

最上边的 `import` 类似 C 语言中的 `#include` 语句，使程序能使用相应类库，能建立 Applet 并获得图形功能的支持。Java 语言为用户提供大量的类库，这些类库以包的形式组织起来，具体内容将在后面的有关章节中加以说明。

当然，用户也可以自己编制各种类库组成包，在程序里加以应用。

为了在 Web 中运行这个小应用程序，必须把它嵌入 HTML(超文本标识语言)的程序中，由 HTML 文件来引导。这可以使用如下一类的标识语句：

```
<APPLET code="HelloWorldB.class" width=300 height=80>
</APPLET>
```

编写 Applet 程序时，必须把程序定义为 Applet 类的子类，这里 HelloWorldB 是父类 Applet 的子类，子类继承了父类的所有方法(method)。方法可以是有某种功能的操作，也可以是施于对象或类上的操作。Applet 使用各种标准方法，这里 `paint()` 作为屏幕显示。

小应用程序也可独立运行，但更多的是在 Web 的客户机/服务器环境运行，在客户机和服务器之间用 HTTP(超文本传输协议)通信，服务器端用 HTML 建立文档页面格式，即 HTML Script 文件，Java 编译器把源程序文件(同样可用各种文本编辑器编辑)，用 `javac` 之类命令进行编译。然后，编译的文件(字节码)在 HTML Script 中被调用。在客户机一端，既需有支持 HTML 语言版本，又要有 Java 字节码解释器才能浏览 Web 页面。Applet 的 Java 字节码是在客户端运行的，因此可以增添浏览器的交互动态特征。浏览器只接受 HTML，故必须建立一个嵌有 Applet 的 HTML 文件。

下面是一个嵌入上述 Applet 程序的由 HTML 语言编写并可实现的简单程序。首先确定该 HTML 语言程序的文件名(例如 HWA.HTML)。则作为相应的简单程序如下：

```
1: <HTML>
2: <BODY>
3: <Applet CODE="Hello WorldB.class" WIDTH=50 HEIGHT=20>
4: </Applet>
5: </BODY>
6: </HTML>
```

这里 CODE 指明字节码所在的编译文件名 HelloWorldB.class。WIDTH 和 HEIGHT 指明 Applet 占有的空间大小。Applet 标记作为指明嵌入 Applet 程序。

浏览 Applet 需应用以下任一种工具：

支持 Java Applet 的浏览器，诸如 Netscape 2.0 以上版本等。

JDK 的 Appletviewer，但 Appletviewer 不是 Web 浏览器，不能看到完整的 Web 页面。

2. Java 虚拟机 (JVM)

Java 是在一个虚拟机的环境下工作和运行的。源程序经过编译后形成字节码文件(.Class)，字节码由不依赖于特定硬件平台的指令所组成。字节码中包含着相应的符号引用信息，而不是把变量和方法的引用编译成数值的引用。然后由解释器在运行过程中安排内存布局，并由查表确定相应的地址。解释器执行分三个阶段：首先是装入代码，由“类载入器”完成。这阶段载入运行时所需的全部代码(包括被调用的类和继承的类)。类之间除了符号引用类名空间外，类之间不再有其他联系方式。在装入类后，解释器就可以确定内存格局。通过查表把符号引用与地址空间对应起来。然后，进行字节码校验，这是由“字节码校验器”

实现的。它检查非法类型转换及栈溢等。最后是代码执行，一般采用解释执行，即每次翻译一段代码，直至完成字节码程序所有的操作。

Java 虚拟机大致分为五个部分：

- 指令集 约有近 250 个指令字节码，每个字节码执行一个基本运算，操作码可以带操作数或不带操作数。

- 寄存器组 主要是用于保存机器运行的信息，不传递和接受参数。它包括：程序计数器 PC；操作数栈端指针 **Optop**；当前执行方法的执行环境指针 **Frame**；指向当前执行方法的局部变量指针 **Var**。

- 栈 包括三个部分：局部变量区，执行环境区和操作数区。

- 无用单元收集堆 主要作为动态分配对象的存储区域。

- 虚拟机的方法区 它是编译后的代码区。

编译后，类作为虚拟机的代码文件仅仅是逻辑上的，虚拟机的内存区不直接依赖于实际的内存区。Java 的虚拟机结构保证了它的高效性和可移植性。

3. Java 语言的开发工具 JDK

JDK (Java Developer's Kit)是开发 Java Applet 和 Application 的工具,大致包括如下的内容: Java 语言编译器(JavaC); 解释器(Java); 文件产生器(Java.DOC); 程序调试器(JDB); 剖析工具(JavaPROF); 类反汇编程序(JavaP); Java Applet 观察器(Appletviewer)及 C 文件生成器(Javah)。

目前 JDK 有 Windows95 版、WindowsNT 版和 Solaris2.X 版等，均可免费获得。SUN 为 JDK 提供两个站点：一个 Web 站点和一个匿名 FTP 站点。可从这些站点下载 JDK 开发工具。

(1) 由 Web 下载 JDK 可在浏览器上输入下列 URL (例如应用 Netscape Navigator 浏览器), <http://java.sun.com/>, 然后, 点击主页上的 Download the jdk1.1 beta!。在下载后的页面上再点击文件 JDK-beta-win32-X86.exe (可修改该文件名), 执行这个自展开文件创建一个 Java 目录及若干子目录, 然后建立相应路径。我们依据实际操作系统可选择合适的 JDK 版本。

(2) 由匿名 FTP 下载 JDK 匿名 FTP 的站点名为 <ftp.javasoft.com>, 登录名用 anonymous, 用自己的电子邮件地址作为口令字。

在不同的操作系统下, 安装的方式不一样。若下载的是 Windows95/NT 的 JDK, 则文件名为 JDK-beta-win32-X86.exe。可把它拷贝到你指定的目录下, 例如, 安装到根目录下, 可执行如下命令

```
C:\temp>copy JDK-beta-win32-X86.exe c:\
```

在 Windows95 中, 可由 Windows Explorer 或从 MS DOS prompt 命令行, 运行 JDK-beta-win32-X86.exe。

它是一个自展开的压缩文件, 它首先将建立一个 Java 目录及其子目录, 然后把 JDK 的所有文件展开到相应目录。在完成后可查看相应内容:

```
C:\JAVA>dir
```

从这一级目录中, 我们可以看到子目录 BIN, 它的目录下含执行文件和动态连接库, CLASSES 目录下保存的是 Java 类库, LIB 目录下保存的是库文件, DEMO 目录下保存的是 Applet 实例。

现在, SUNSoft 公司新推出了 Java Workshop Develop Products。它是第一个用于创建

Internet 应用程序和 Web 页的完整的集成开发环境。它除了包括集成的窗口式开发调试工具外，还有使用方便的图形 Web 浏览器以及在 Internet 上公布的 Java 应用程序工具。

4. HotJava

它是用 java 语言编写的，是一个新型的 Web 浏览器，也是基于客户机/服务器体系结构方式，但把交互中心由 Web 服务器转移到 Web 客户机。它把 HTTP 作为主要的通信协议，并可动态地下载程序代码在客户机上执行。可运行“小应用程序” Applet，故可以提供实时交互和图像、声音及动画功能。而 Mosaic 等浏览器只能静态地浏览数据，不能充分地使用网上资源，不能与服务器交互操作。

HotJava 所理解的文件类型是 HTML(超文本标识语言)，它在 HTML 语言中新增加了 <APP> 标记，当 HotJava 浏览器遇到 <APP> 标记时便向服务器请求相应的程序，并解释执行这些程序。

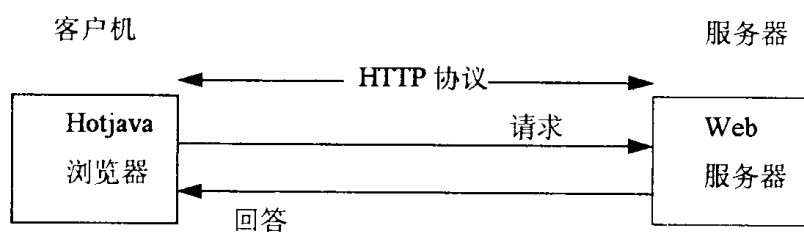


图 1-1 基于 HTTP 协议的浏览器和服务器间通信

HotJava 的主要特点如下：

交互能力强： 可用 HotJava API(应用程序接口，给出用户应用程序所需的一些函数和变量)编写相应的 Java 程序实现交互的页面。在 Alpha3 版本中有排序算法(冒泡，双向冒泡，快速排序等)的一些生动实例。当数据直方图被单击后，这些代表数据的参差的直线最终通过反复交换后，表现为有序排列。同样可以通过单击一些图面或算法，来观看仿真实验过程和算法演示。

动态地处理格式及类型： 一般浏览器只能显示 GIF 等格式的图片，对于新的格式就不能适从。HotJava 能动态地从安装了处理代码的服务器下载处理新格式代码(HotJava 浏览器会自动向服务器请求)。这个过程对用户是透明的。

动态的协议支持： HotJava 采用外置动态协议，只需把协议的处理程序挂在浏览器上，用户便可使用协议名字(在 URL 中写上协议名)来链接相应处理程序。

支持产品的升级换代： 以往，一旦使用新协议和数据类型，就必须更新所有客户机上的浏览器；而 HotJava 使新的数据类型连同它的支持软件，在服务器上安装之后，产品的更新便自动完成。

安全性： HotJava 可以让用户规定应用程序访问 URL 类的权限，从而提高了抗病毒的能力和安全性。

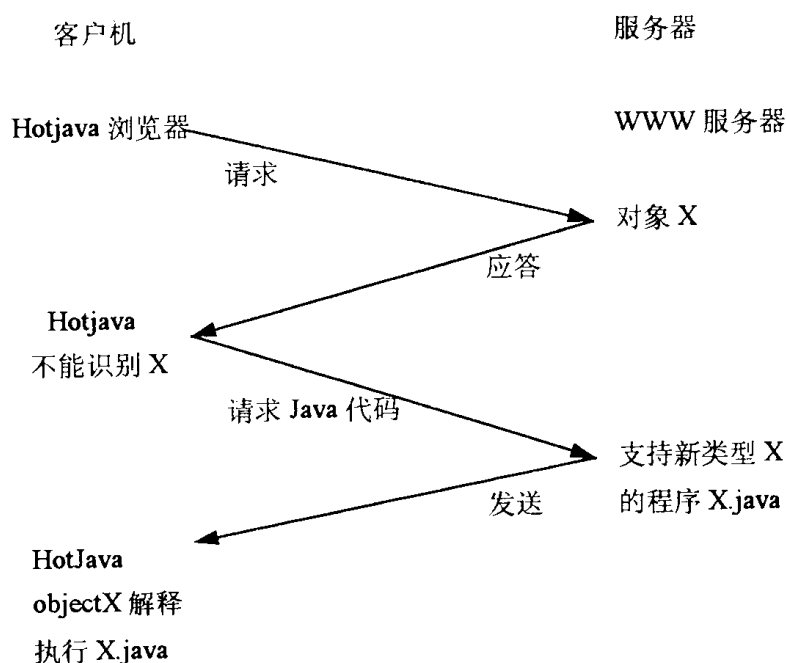


图 1-2 HotJava 支持动态可扩充数据类型

1.2 面向对象编程及 Java 语言的特点

1. 面向对象编程简介

客观世界的任何事物都可视为对象。我们把数据及其操作方法与对象对应起来，使对象成为封装的数据类型和相关的操作，并以此作为编程的基本出发点，改变了过去把二者分离的编程方式，使它更切合现实世界模式。在面向对象编程中，我们命名“类”(class)作为具有统一属性和操作方式的一群对象描述体，即建立在支持数据的抽象及实现对数据约束的基础上。这里属性是指描述对象的一组数据，是表示为对象的一些变量，而用操作代码表示对象的行为。例如我们声明自行车对象，它的属性有颜色、车身尺寸、材料、商标等。而对它的操作有运动、修理、停靠等。这些就形成了自行车一般的“类”。尽管有的类并不表示物理对象，但它们作为概念上的实体，可用相关用语来表明。

所以，类仅是基于对象之上的一个概念，是创建对象的模板。我们把“实例”(instance)作为对象的具体表示。对于被创建的对象，也被称为类的一个实例。例如自行车类中最后确定为：颜色是红、车身为 26 寸、商标是“凤凰”等。对象被实例化后，方法将顺次执行每个语句。

2. 面向对象的一些基本概念

在面向对象编程中，我们并不首先具体地定义对象，而是定义对象的类。把对象作为类的实例。将相关的数据及其操作组织在类内，构成具有独立意义的构件，称为“封装”(Encapsulation)。它把一些关系隐藏在内部，对外仅表现为封装体间的接口关系。类也是构成程序的很好模块，它满足了模块原则及标准，并能满足代码复用的要求。

除了封装外，面向对象编程语言还具有其他的一些重要技术，它们是“继承”(Inheritance)和“多态性”(Polymorphism)等。继承是一个类直接使用另一类的所有属性和方法。它可以减