

网络编程语言

Java

孙淑玲 王太权 陈意云 编著



中国科学技术大学出版社

7.912
C.35

网络编程语言 Java

孙淑玲 王太权 陈意云 编著

中国科学技术大学出版社

合肥·1997

内 容 简 介

Java 语言是面向对象的网络编程语言，已成为软件开发的一种革命性技术。它以简洁、平台无关、多线程、高安全性为特征，确立了在网络计算时代中的地位。本书介绍了 Java 语言的基本程序设计结构，详细叙述了面向对象程序设计的对象、类、程序包、界面等概念；以及用 AWT 进行图形和用户界面的设计；异常处理；输入输出；多线程和网络连接。书中配有大量典型示例，适合学习和使用 Java 语言的各层次人员，特别是大学生和软件开发人员。

图书在版编目 (CIP) 数据

JS108/12

网络编程语言 Java / 孙淑玲等 编著

——合肥：中国科学技术大学出版社，1997 年 8 月

ISBN 7-312-00881-X

I. 网 络 编 程 语 言 Java

II. 孙淑玲 王太权 陈意云

III. ①网络 ② Java 语言 ③程序设计

IV. TP

中国科学技术大学出版社出版发行

(安徽省合肥市金寨路 96 号, 邮编 230026)

安徽合肥骆岗印刷总厂印刷

全国新华书店经销

开本：787×1029/16 印张：22 字数：520 千字

1997 年 8 月第 1 版 1997 年 8 月第 1 次印刷

印数：1—5,000 册

ISBN 7-312-00881-X/TP · 172

定价：29.00 元

前 言

Internet 已被公认为本世纪末人类科技史上的里程碑。Internet 的应用范围迅速从传统的科研、教育领域扩展到商业、办公领域和家庭环境,促使人类社会步入以网络为中心的信息时代。

以往我们开发的软件编译后就被约束在特定的硬件平台、操作系统以及网络环境,软件的移植性较差。在以网络为中心的时代,需要一种独立于平台、代码可移植的计算技术。Java 语言恰恰以其面向对象、动态交互操作与控制、动画显示、不受平台制约等优势填补了这个空白。从 Java 诞生之日起便受到众多著名计算机厂家和全球上千万用户的青睐。

目前 Java 已是广泛使用的网络程序设计语言,它的设计思想是最大限度地利用网络。它体现一种全新的计算概念,并被视为通向网络计算时代的入场券。Java 语言的面向对象、完全的指针控制、无用单元自动回收、多线程多任务、平台无关性和高安全性,使它成为今天软件开发的一种革命性技术。它的出现将加速应用软件小型化和网络化趋势。Java 提供的代码可移植性和平台无关性,使网络交换的不仅是信息,还可以是程序实体。所以它也是很好的分布式对象技术。现在在许多商业领域已开始用 Java 编写程序,包括安全性要求极高的金融服务;例如,在零售业和广告可以用 Java 来做交互购物、现场拍卖和交互式 3D 广告;在出版业和娱乐业用 Java 安全地在 Internet 上发布电子出版物、实时信息和多用户游戏;在信息管理系统可以用 Java 来实现关键任务应用程序和网络数据库访问;教育工作者可用 Java 来实现远程学习和交互式模拟。

1996 年 5 月,世界 Internet 和 Apple、IBM、Netscape、Oracle、Sun 五家公司连联合发布了网络计算机标准规范,它标志着网络计算机的正式诞生。Java 是网络计算机的灵魂和强有力的武器。不难看出,Java 语言必将对全球计算机产业产生深刻影响,对传统的计算模型提出新的挑战。

我国信息化建设正在高速发展,网络化热潮正席卷各行各业。及时地掌握和研究 Java 语言,推进我国信息产业的发展,以便在新的竞争中处于有利地位。这也是我们写这本书的初衷。

本书的内容安排是:在介绍 Java 程序设计结构的基础上,引入对象、类及其继承性等面向对象程序设计概念;讲述如何在没有指针的情况下实现链表、栈等复杂数据结构;详细叙述如何使用 AWT 编写图形程序和用户界面;接下又引入了异常和多线程概念,并讲述输入输出流和网络连接。其中第 1, 2, 3, 4 章由陈意云和章洁编写、第 5, 6, 7 章由张克和马海寅编写、第 8, 9, 10, 11 章及附录由孙淑玲和王太权编写。书中配有大量典型实例及实例分析。

本书读者最好具有基本的 C 语言程序设计基础。

编者

1997 年 1 月

目 录

第一章 Java 语言简介	1
1.1 Java 语言的背景	1
1.2 Java 语言的特点	2
1.3 Java 与 Internet	4
1.4 Java 对软件开发技术的影响和应用前景	6
第二章 Java 的基本程序设计结构	8
2.1 一个简单的 Java 程序	8
2.2 数据类型和变量	9
2.2.1 注释	9
2.2.2 标识符	10
2.2.3 关键字	10
2.2.4 字面常量(Literal)	11
2.2.5 简单类型	13
2.2.6 变量和常量	15
2.2.7 复合类型	16
2.3 运算符	20
2.4 语句	21
2.4.1 声明	21
2.4.2 表达式	21
2.4.3 控制流	22
2.5 实例:分期付款计算器(Mortgage Calculator)	24
2.6 类方法	26
2.7 递归(Recursion)	28
第三章 类和对象	30
3.1 面向对象程序设计的基本知识	30
3.1.1 OOP 的出现	30
3.1.2 OOP 的基本概念	30
3.1.3 对象	31
3.1.4 类间的关系	32
3.1.5 OOP 与传统的面向过程的程序设计的比较	33
3.2 使用已有的类	34
3.2.1 对象变量	34
3.2.2 类 Date	36
3.2.3 类 Day	37

■

3.3 创建自己的类.....	43
3.3.1 类 Employee	43
3.3.2 类 CardDeck	53
3.4 包(package)	57
3.4.1 包的组织和命名.....	57
3.4.2 包的使用.....	58
3.4.3 包的定位.....	58
3.4.4 包的作用域.....	59
3.5 类的设计提示.....	59
第四章 继承(inheritance)	62
4.1 继承.....	62
4.1.1 类 Manager	62
4.1.2 子类的使用.....	67
4.1.3 多态性.....	67
4.1.4 final 类	68
4.2 强制类型转换(Casting)	69
4.3 抽象类(Abstract Classes)	70
4.4 界面.....	76
4.4.1 界面的定义.....	76
4.4.2 实现多个界面.....	82
4.4.3 扩展界面.....	83
4.4.4 界面和回调.....	83
4.5 广义超类(Cosmic Superclass)	85
4.5.1 类 Object	85
4.5.2 对象包装类(Object Wrappers)	86
4.5.3 阅读 API 文档	88
4.6 类 Class(运行时的类型识别信息)	90
4.7 protected 访问	91
4.8 几个复杂的数据结构.....	94
4.8.1 链表.....	94
4.8.2 队列	100
4.8.3 向量(vectors)	100
4.8.4 栈	106
第五章 Java 的图形编程	108
5.1 AWT 概述	108
5.2 在窗口中输出文本	110
5.2.1 输出简单文本	110
5.2.2 字体和颜色	112
5.3 图形图像的处理	118

5.3.1 图形的绘制和填充	118
5.3.2 绘图模式	122
5.4 图像处理	124
第六章 用户图形界面设计	131
6.1 概述	131
6.2 面板和画布	132
6.3 文本处理	135
6.3.1 文本输入	135
6.3.2 有效性检查	138
6.4 选择的方式	141
6.4.1 检查框	141
6.4.2 检查框组	142
6.4.3 选择框	142
6.4.4 列表	143
6.5 组件的排列方式	146
6.5.1 顺序排列方式	146
6.5.2 分区排列方式	146
6.5.3 卡片排列方式	147
6.5.4 网状排列	148
6.5.5 高级网状排列	151
6.5.6 任意排列	153
6.5.7 自定义排列方式	153
6.6 对话框	157
6.7 数据交换	159
6.7.1 数据交换	159
6.7.2 文件对话框	163
6.8 AWT 的事件处理	164
6.8.1 概述	164
6.8.2 键盘事件处理	167
6.8.3 鼠标事件处理	167
6.9 菜单设计	171
6.10 滚动条的实现	172
6.10.1 作为滑动控制组件	173
6.10.2 滚动窗口区域	175
第七章 小应用程序的设计	179
7.1 小应用程序的基本概念	179
7.1.1 小应用程序和 HTML	179
7.1.2 一个例子	179
7.1.3 小应用程序的执行方式	180

7.1.4	安全限制	180
7.2	应用程序到小应用程序的转换	181
7.3	HTML 中与小应用程序有关的标记及属性	185
7.3.1	与小应用程序有关的标记	185
7.3.2	与小应用程序相关的 HTML 属性	185
7.4	小应用程序的参数传递	193
7.5	小应用程序中的对话框	194
7.6	多媒体	195
7.6.1	什么是 URL	195
7.6.2	获取多媒体文件	196
7.7	小应用程序的执行环境	197
7.7.1	小应用程序的通信	197
7.7.2	利用浏览器显示信息	198
7.7.3	Netscape 的框架及其使用	199
7.8	小应用程序的生存期	202
7.8.1	概述	202
7.8.2	一个网络漫游的实例	203
7.9	既是小应用程序又是应用程序的 Java 程序	208
第八章	异常处理	212
8.1	异常概念	212
8.1.1	异常	212
8.1.2	异常的层次结构	212
8.2	异常的引发	215
8.3	异常的捕捉与处理	215
8.4	异常的传播	217
第九章	输入和输出	220
9.1	文件和标准类型	221
9.1.1	文件	221
9.1.2	标准流	224
9.2	输入输出流	225
9.2.1	打印流	226
9.2.2	文件输入和输出	227
9.2.3	过滤输入输出流	229
9.2.4	有管道的输入流和输出流	234
9.2.5	序列输入流	241
9.2.6	字符串缓冲区输入流	242
9.2.7	字节数组输入输出流	243
9.2.8	随机访问文件流	244
第十章	多线程	248

10.1	序言	248
10.2	什么是线程	248
10.2.1	线程的描述	252
10.2.2	线程的启动和运行	253
10.2.3	多个线程同时运行	256
10.3	线程的特性	257
10.3.1	线程的状态	257
10.3.2	线程状态的检测	259
10.4	线程的优先级	260
10.5	线程间的相互作用	264
10.5.1	线程组	265
10.5.2	线程间的同步	266
10.6	基于线程的程序设计	274
10.6.1	计时器	274
10.6.2	哲学家问题	278
10.6.3	动画	280
第十一章	网络程序设计	287
11.1	引言	287
11.2	利用 URL 实现网络通信	287
11.2.1	什么是 URL	287
11.2.2	从一个 URL 地址读取资源的内容	287
11.2.3	向一个 URL 地址写入信息	289
11.2.4	从远程结点检索信息	291
11.3	基于数据报的网络程序设计	296
11.3.1	什么是数据报	297
11.3.2	一个数据报服务程序	297
11.3.3	一个数据报客户程序	302
11.4	基于套接字的网络程序设计	304
11.4.1	基于套接字的网络通信	304
11.4.2	客户程序的实现	305
11.4.3	服务器的实现	307
11.4.4	向服务器发送信息	313
附录 1	Unicode 标准	321
附录 2	错误与异常类	323
附录 3	内部定义的方法	325
	参考文献	339

第一章 Java 语言简介

计算机经过 50 年的发展, 已经经历了以大型机为中心的集中处理时代和以个人机为中心的分散处理时代, 现在正进入网络时代。网络时代的特点是各种硬件和软件资源都尽可能地放在网络中, 供大家共享, 各种处理也尽可能地由网络服务器进行。要使网络时代成为现实, 还要人们能够自由地使用网络中的软件。由美国 Sun Microsystem 公司开发的新一代程序设计语言 Java, 便是面向这个目标的工具。

1.1 Java 语言的背景

Java 语言的开发始于 1991 年。其开发小组, 包括 Java 的创始人 James Gosling, 当初的目的是开发简单且无错的消费类电子产品。在研究开发过程中, Gosling 深刻认识到消费类电子产品和 workstation 产品开发原理之间的差异。消费类电子产品要求可靠性高、费用低、标准化、使用简单, 而 workstation 用户则要求机器有强大的计算能力, 他们并不在乎价格的高低和操作的复杂性。消费类电子产品用户并不关心 CPU 的型号, 也不欣赏专用且昂贵的 RISC 处理器。因此对开发者来说, 需要找到一种办法, 建立独立于平台的代码, 即让该软件在任何的 CPU 上都能运行。为达此目的, Gosling 首先从改写 C++ 编译器着手, 但他很快发现仅是 C++ 还无法满足需求, 于是在 1991 年 6 月开始准备开发一种新语言。这种语言最初被称为 Oak(一种橡树的名字), 后发现 Oak 是 Sun 公司另一种编程语言的名字, 于是改名为 Java。

Java 走红于 1995 年和 1996 年。1994 年下半年, Internet 的迅猛发展, 环球网 WWW 的快速增长, 促进了 Java 语言开发的进展, 使得它逐步成为 Internet 上颇受欢迎的开发和编程语言。一些著名的计算机公司纷纷购买了 Java 语言的使用权, 如 MicroSoft、IBM、Netscape、Novell、Apple、DEC 和 SGI 等。Java 语言被美国的著名杂志 PC Magazine 评为 1995 年十大优秀科技产品之一。1996 年 5 月 29—31 日, 在美国旧金山召开了世界 Java 语言开发者大会, 与会人数多达 6500, 来自工业界的人超过一半。工业界一致认为“Java 是 80 年代以来计算机界的一件大事”。微软总裁比尔·盖茨认为“Java 是长期以来最卓越的程序设计语言”。Sun 公司总裁 Scott McNealy 认为 Java 与 Internet 和 WWW 开辟了一个崭新的时代。WWW 的创始人 Berners-Lee 说: “计算机事业发展的下一个浪潮就是 Java, 该浪潮很快会发生”。就连华尔街金融界也投入资金、人力用 Java 开发电子贸易、金融软件。随着 Internet 在中国的逐步普及, 1996 年春, Java 随着 WWW 之风吹遍神州大地。

Java 语言的使用已成大势所趋。MicroSoft 和 IBM 两大公司都计划在 Internet 上销售用 Java 编写的软件。Apple、HP、IBM、MicroSoft、Novell、SGI 和 SCO 等公司均计划将 Java 虚拟机和 Applets 类库嵌入各自开发的操作系统, 这样各类开发人员就能更容易地选

择多种平台来使用 Java 语言编程,不同的用户也就可以脱离 WWW 浏览器来运行 Java 应用程序。这无疑是受广大用户欢迎的,也为 Java 语言的应用开拓了极为广阔的前景。另外,由 JavaSoft 推出的完全用 Java 编写的 Internet 上新型浏览器 HotJava,不仅能编制动态的应用程序,而且能编制完整的成套桌面应用软件。这也为 Java 的应用提供了有力的例证。

MicroSoft 整个软件开发战略将从 PC 单机时代向着以网络为中心的计算时代转移。有人预言,Java 语言将成为网络上的“世界语”,今后所有用其它语言编写的软件都要用 Java 语言来改写。全世界都对 Java 寄予厚望。

1.2 Java 语言的特点

Java 语言是一种广泛使用的网络编程语言,它最大限度地利用了网络,Java 的小应用程序(applet)可以在网络上传输而不受 CPU 和环境的限制。Java 又是一种面向对象的编程语言,支持面向对象的程序设计方法,用它可以建立可复用的软件部件。

Java 语言有下面一些特点:简单性、面向对象、分布式、健壮性、安全性、体系结构中立、可移植性、解释执行、多线程性和动态性。

1. 简单性

这是 Java 语言的最重要的特点。Java 与 C++极为相似,但相对来说要简单得多,它删去了 C++中几乎不用或很难用好的一些语言特征。例如,它不支持 struct、union 和指针数据类型,不支持 typedef、#define 和#include, 不支持运算符重载、多继承性和虚基类。Java 还提供了一个无用单元收集器,从而减轻了程序员进行内存管理的负担,避免犯一些不该犯的错误。

2. 面向对象

Java 是一种面向对象的编程语言,除了基本类型,如数值型和布尔型以外,其余大部分都是类。Java 的代码也是按类组织的。类是 Java 应用程序和小应用程序的构造块,程序的所有部分都必须在类中存在。每个类定义一组属性和一组规定对象行为的方法。一个类可以继承另一个类的属性和行为。在类的根层次上,通常是类 Object。

Java 只支持单继承的类层次结构,即每个类只能继承一个类。而 C++却支持多重继承。类的多重继承很可能造成混乱,使语言本身及其实现复杂化。例如,难以想象一个对象继承两个完全不同的类的行为。

Java 提出了“界面(interface)”概念,界面只定义了一组方法而没有实现其体,即界面只提供方法协议的封装,而不限制实现必须在什么继承树上。当一个类实现界面时,一般必须实现界面描述的所有方法的体。

一个类可以实现多个界面,由此实现多继承性的许多优点。一个对象也可实现任意数目的界面。Java 界面与 IDL 接口十分相似。很容易建立一个 IDL 到 Java 的编译程序,即很容易建立 IDLJ 编译器。这就是说,Java 可用于 CORBA 对象系统来建立分布式对象系统。鉴于在许多计算机系统中都采用 IDL 接口和 CORBA 对象系统,这种兼容性是重要的。

3. 分布式

Java 有一个内容广泛的网络类库,它是对分布式编程的最好支持。这个网络类库是支持 TCP/IP 协议的子例程序,目前支持的协议有 HTTP 和 FTP 等。Java 应用程序可以通过

网络,借助于 URL 来打开和访问对象,这和程序访问本地文件系统一样容易。

Java 的网络特性既强又易使用,任何一个想利用 Internet 完成程序设计的人都将倾向于用 Java,因为 Java 可以使一些复杂的任务,象打开一个套接字(socket)连接等任务简单化。Java 甚至可以使公共网关接口(CGI) scripting 更加简单。

4. 健壮性

Java 强调对可能出现问题的早期检查和运行时的动态检查,避开一些易犯错误的场合。Java 编译器可以检查很多错误,而这些错误在别的语言中是放在运行时检查或者根本不检查。Java 不提供指针类型,但它有一个指针模型。你不需要用指针来表示常用结构如字符串数组等,但当需要指针时,例如链表,它又有指针的功能。这就排除了程序员误写内存和破坏内存的可能性,消除了程序员在分配和释放内存时可能发生的误操作。这个特征将为那些花费数小时寻找因指针错误引起内存丢失的程序员提供了方便。

5. 安全性

要在 Internet 上实现安全性,有一条严格的规则:不下载任何可执行的代码,除非这些代码来自于你所熟悉和可以信赖的人员或公司。

可喜的是,Java 允许人们放宽该规则,人们可以十分安全地执行来自任何地方和任何人用 Java 编写的应用程序。

Java 的坚强的安全机制表现在系统体系结构的四个不同的级别上。第一,Java 语言本身的设计考虑了安全性,Java 编译程序保证源代码不违反这些安全规则。第二,Java 运行时系统所执行的所有字节代码经过了检验以保证它们也遵守这些规则;第三,类装入程序在把类装入到系统时保证类不侵犯名字空间或访问限制的范围。第四,API 专用安全性防止 Java 应用程序的破坏性操作。最后一级取决于其它三级的安全性和完整性。

6. 体系结构中立性

Java 编译器产生一种体系结构中立的文件格式,即编译得到的代码在许多处理器上都可以执行,当然这些处理器上必须存在 Java 运行系统。Java 编译器是通过产生与任何特殊计算机体系结构无关的字节代码指令来实现上述功能。字节代码指令在任何机器上都容易被解释,也很容易翻译成实际的本地机器指令,这说明 Java 程序的移植很容易,不需要进行重新编译。

7. 可移植性

Java 不依赖于实现,这一点与 C 和 C++不同,Java 的基本数据类型的大小是固定的,例如: int 在 Java 中总是 32 位,在 C/C++中, int 可以认为是 16 位整数、32 位整数或者任何语言实现者喜欢的某个大小的整数。固定数值类型的大小剔除了可移植性的主要难题。Java 中二进制数用固定格式存储,字符串用标准的 Unicode 风格存储,这些也增强了可移植性。

作为系统一部分的库也定义了可移植的界面,例如:库中存在一个抽象类 Window 和它在 Unix、Windows 和 Macintosh 下的实现。

8. 解释执行

Java 解释器可以直接执行 Java 字节代码,而且由于链接是一个增量的、低代价过程,所以开发过程会更快、更具有探索性。但是,由于解释代码的速度比编译代码的速度慢(慢 10 倍),人们最终还是想要一个编译器。

9. 多线程

Java 程序可以执行一个以上的线程。比如它可以在一个线程中完成某一耗时的计算，而在其它线程中与用户进行交互对话，所以用户不必为等待 Java 程序耗时的计算而停止工作。

多线程在 Java 中实现比在其它语言中实现来得简单。虽然对不同的平台，线程的实现大不相同，但是 Java 把多线程的实现交给了底层的操作系统，清楚地向上体现出平台无关性。因此，如果底层的操作系统能够利用硬件的多处理机特性的话，那么 Java 的多线程也能利用这一点。

10. 动态性

在很多方面，Java 是一种比 C 和 C++ 更具动态性的语言。它被设计成一种适应不断变化环境的结构。在 C++ 中，如果父类改变，那么往往需要彻底重新编译；而 Java 使用界面的方法减轻了这种依赖。结果是，Java 程序允许新的方法和实例变量加入对象库中而不影响它们的应用程序。此外，Java 中可以直接找到运行时的类型信息，在要求向正在运行的程序中加入代码的情况下，这个特征显得十分重要。另一方面，Java 提供的运行时的类型信息比 C++ 提供的要丰富得多。

1.3 Java 与 Internet

Java 不仅代表由 SUN 公司开发的一种编程语言，也代表用这种语言去提高 WWW 上交互性的一组软件工具。Java 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这种软件的平台。这种软件可以支持客户机端的交互活动，使 WWW 的交互性升到一个前所未有的水平。

作为 Internet 上的一个客户机/服务器方式的应用系统，WWW 使用户可以通过超文本链在网上选择所需的多媒体信息，但是这种交互性非常低。公共网关界面 CGI 编程提供了高一级的交互性：服务器将客户机送来的请求转交给 CGI 程序，CGI 程序负责信息检索或其它处理，并将结果以 WWW 页面的形式返回给客户机。

Java 除了支持上述功能外，还通过一种特定的 HTML 标记 APPLET，使开发者能够把 Java 小应用程序包含在 WWW 页面中。这种小应用程序由用户的 WWW 浏览器取回并且执行。Java 通过小应用程序进行的各种输入和输出，使 WWW 的交互性更加丰富多彩。

下面是一个最简单的 Java 程序。首先建立一个 HTML 文件 HelloWorld.html 如下：

```
<HTML>
<HEAD>
  <TITLE>First Hello Applet</TITLE>
</HEAD>
<BODY>
  <P> "This is it!" <APPLET code= "HelloWorld.class" >
</BODY>
</HTML>
```

在这个文件中的 APPLET 标记指示一个叫做 HelloWorld 的 Java 小应用程序，它将包

含在 WWW 的页面中。

然后我们来编这个Java程序HelloWorld.java, 和HelloWorld.html放于同一目录下。

```
import browser.Applet;
import awt.Graphics;
public class HelloWorld extends Applet
{
    public void init()
    {
        resize(600, 300);
    }
    public void paint(Graphics g){
        g.drawString( "Hello World!", 50, 100);
    }
}
```

这个程序经编译器

编译生成HelloWorld.class, 它包含了HelloWorld 小应用程序的字节代码。编译好后, 把HelloWorld.class放到当前目录下。

如果你启动一个Java浏览器, 如HotJava, 那么可以测试这个小应用程序。首先用浏览器打开服务器上HelloWorld.html。然后浏览器根据该文件中的APPLET标记, 从服务器中将HelloWorld.class取来, 并调用Java解释器对其字节代码进行解释执行, 在页面(50, 100)处写出Hello World!。图1.1概括了这个过程。

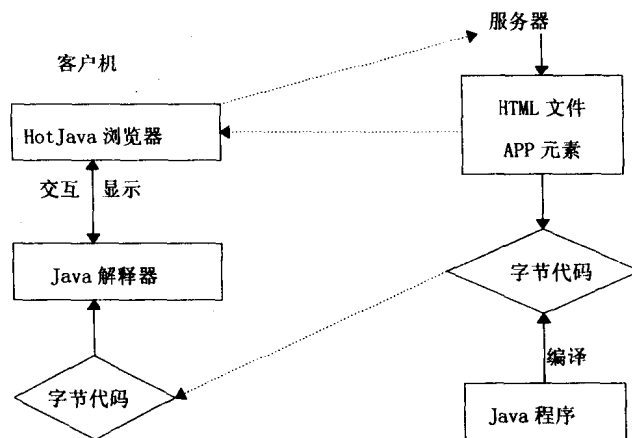


图 1.1 Java 的执行过程

Java 有足够的力量, 浏览器 HotJava 本身就是 Java 的一个应用程序。Java 对 WWW 网上的交流起着一种变革的作用。

1. Java 可使 WWW 页面栩栩如生

在 Java 诞生前, 浏览器的超文本页面只能显示静止图像, 在其它有关程序的支持下方可显示视频信息。使用 Java 浏览器可使 WWW 页面生气勃勃, 不仅可有活动的图像和

正文，还可以包含程序员可想象出的任何活动的可视元素。

例如，Java 可使正文闪烁、快速滚动等。再例如，可以在页面显示两只手扔几个球的杂技动作。在一般的动画技术中，要用不同的帧来表示小球在它路径上的不同位置。而在 Java 中是把小球的路径存入一个数组，沿着路径重新定位小球的图像。在可点图方面，用 Java 可以实现这样的功能：当光标移到图像的热点区域时，这个区域会加亮，点击这个区域可取回别的资源。

在进行开发时，你可以修改和使用网上其它站点的开发者的小应用程序，就象把超文本页面链到其它站点一样。

2. Java 使 WWW 页面的交互性更好

Java 不仅能使 WWW 页面活起来，而且 Java 的小应用程序可以接收来自鼠标和键盘的用户输入。用户的输入可以影响小应用程序的信息处理流程。因此可用 Java 来编写交互式的游戏程序和教育软件，如通过键盘输入来改变图形显示的游戏程序。

3. Java 的网络特征

Java 浏览器的一个最有意义的特征是它不局限于已经掌握的协议和媒体格式，它可以调整到学会新的协议和媒体格式，只要把新的协议处理器和内容处理器编译后放到浏览器文件目录结构的恰当位置并通知浏览器就行了。这就扩大了 Java 浏览器能解释和显示的范围。

Java 还有这样的能力，它能把网络信息的收取放在小应用程序中。例如，可以编写这样的网络导游小应用程序，它从你选择的某个 URL 开始，然后接管你的浏览器，每隔一段时间从一张 URL 表中随机选取一个 URL 并显示相应的新页面。这在教育和广告等方面有用处。

Java 可在网上传递可执行的内容，即字节代码。

1.4 Java 对软件开发技术的影响和应用前景

不少人预言：“Java 的出现会引起一场软件变革”，这是因为传统的软件往往都是与具体的实现环境有关，换了一个环境就需要作一番改动，耗时费力，而 Java 语言能在字节代码上兼容，这样所开发的软件就能运行在不同的机器上，只要所用的机器能提供 Java 语言解释器即可。

Java 语言将对未来软件的开发产生深远的影响，可概括为如下五个方面：

- 可将用户的需求进行动态的、可视化描述，提供给设计者更加直观的要求。而用户的需求是各色各样的，不受地区、行业、部门和爱好的影响，都可以用 Java 语言描述清楚。
- 由于 Java 语言的面向对象的特性，所以完全可以用面向对象的技术与方法来开发软件。这对面向对象方法的推广无疑会有巨大作用。
- Java 语言的动画效果远比 GUI 技术更加逼真，尤其是利用 WWW 提供的巨大动画资源空间，可以共享全世界的动态画面的资源。
- 用 Java 语言开发的软件可以具有可视化、可听化和可操作化的效果，这要比电视、电影的效果更为理想，因为它可以做到“即时、交互、动画与动作”，要它

停就停，要它继续就继续，而这在电影与电视播放过程中是难以做到的。

- 使用 Java 语言对开发效益、开发价值都有比较明显的影响。

Java 有着广阔的应用前景，目前至少可以看出有以下几个方面：

- 所有面向对象的应用开发，包括面向对象的事件描述、处理和综合等；
- 计算过程的可视化、可操作化的软件的开发；
- 动态画面的设计，包括图形、图像的调用；
- 交互操作的设计(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 WWW 页面的动态设计、管理和交互操作设计等；
- Intranet(企业内部网)上软件开发(直接面向企业内部用户的软件)；
- 与各类数据库连接查询的 SQL 语句实现。

第二章 Java 的基本程序设计结构

上一章简单介绍了 Java 的一些基本知识,从这一章开始将陆续介绍 Java 编程的工具和方法。

本章主要介绍 Java 程序设计的一些基本概念,比如:数据类型、循环等等;还将简单介绍一下类方法 (Class Methods),也称用户自定义函数。

由于 Java 程序的输入和输出比较复杂,所以在本章中列举的所有例子都是“玩具”程序,也就是说这些例子仅是为了说明概念而编写的。所有的例子都是简单地从控制台(Console)接收数据或往控制台发送数据。有关输入输出的基本操作将在本章和下一章陆续介绍。

2.1 一个简单的 Java 程序

为了便于对 Java 程序设计语言的理解,先来看一个简单的 Java 程序,程序只是简单地往控制台送出一字符串。

例 2.1

一个简单的 Java 程序:

```
1    public class FirstSample
2    {
3        public static void main(String[] args)
4        {
5            System.out.println("We will not use 'Hello World!'");
6        }
7    }
```

例 2.1 程序给出了一个 Java 程序最基本的组成部分。Java 语言的源程序代码由一个或多个文件组成,除了注释和空格之外,每个文件只能包含下列内容:

- 程序包语句
- 引入语句
- 类的声明
- 界面声明

例 2.1 程序由一个文件组成,这个文件只含类的声明部分。由于 Java 程序的所有部分都必须存在于类中,所以文件中类的声明是不可省的。例 2.1 程序声明了一个类 FirstSample。“FirstSample”是类名,通常可以把类看成是一个“盛放”数据和方法(Methods)的“容器”,也就是像第一章所说的:“类是 Java 应用程序和小应用程序的构造块,Java