

新编计算机类本科规划教材

信息与电子学科百本精品教材工程

新编计算机类本科规划教材

Java程序设计 实用教程

朱战立 沈伟 编著

<http://www.phei.com.cn>



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

新编计算机类本科规划教材

Java 程序设计实用教程

朱战立 沈 伟 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

Java 语言是目前最流行,也是最有前途的面向对象程序设计语言。本书将 Java 语言和面向对象程序设计方法相结合,以大量实例详细介绍 Java 的编程思想和编程方法。全书共分 13 章,主要内容包括:Java 语言基础、类、对象、继承、多态、接口、包、Java API 基础、图形用户界面、Java 小程序、异常处理、输入/输出流、多线程、数据库应用、网络通信和 JSP 简介。每章后都设计了大量的基本概念题和程序设计题。全书贯彻实用型教材的编写原则,概念叙述深入浅出,知识点结合实例讨论。

本书既可作为高等院校 Java 语言程序设计课程的教材,也可作为从事软件设计的工程技术人员的技术参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java 程序设计实用教程/朱战立,沈伟编著. —北京:电子工业出版社,2005.1

新编计算机类本科规划教材

ISBN 7-121-00715-0

I. J… II. ①朱…②沈… III. Java 语言—程序设计—高等学校—教材 IV.TP312

中国版本图书馆 CIP 数据核字(2004)第 133361 号

责任编辑:李岩

印 刷:北京大中印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:18.25 字数:467 千字

印 次:2005 年 1 月第 1 次印刷

印 数:5 000 册 定价:24.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010) 68279077。质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

随着计算机网络的飞速发展,网站建设和基于网络的应用程序开发的需求越来越大。用户已经使用多年的许多计算机应用软件也在重新改写为基于网络的版本。面对网络化应用程序的巨大需求,选用哪种程序设计语言来进行开发呢?Java 语言是目前最佳的一种选择。

Java 语言是一种纯面向对象程序设计语言。Java 语言的支持网络应用编程、可跨平台使用、安全性好、支持线程功能等特点,使它成为非常适合网络应用程序开发的一种程序设计语言。

目前,高等学校计算机学科的所有本科专业基本上都开设了 Java 语言程序设计课程,其他许多学科或专业,如通信、仪器等,也都开设了 Java 语言程序设计课程;但根据作者的教学经验,实用类型的 Java 程序设计教材还比较缺乏。

本书的编写方针是:理论叙述深入浅出,设计举例丰富多样,设计要点结合实际例子给出。作者在编写过程中,概念介绍时力求文字准确简洁,设计举例时详细讨论设计要点。书中稍微复杂一些的设计举例都包括题目和要求,设计分析,程序代码,程序设计说明,程序运行结果及运行结果说明。另外,本书还介绍了 JDK 开发环境和 Tomcat 服务器的安装和设置方法,以及 Access 数据库和应用程序建立连接的方法。作者相信,这样的一本教材一定能受到教师和学生的欢迎。

本书内容丰富全面,全书共分 13 章,主要内容包括 Java 语言基础、类、对象、继承、多态、接口、包、Java API 基础、图形用户界面、Java 小程序、异常处理、输入/输出流、多线程、数据库应用、网络通信和 JSP 简介。这些内容可以满足大部分学校的教学要求。作者在每章后设计了大量的基本概念题和程序设计题。这些习题可以帮助学生巩固概念,并真正具备动手设计能力。

本书的所有例子都上机调试通过。

根据作者的教学体会,使用本教材授课约需 40~54 学时。课时不多时,后边几章的内容可根据情况少讲或不讲。

本书第 1~第 5 由朱战立编写,第 6~第 13 由沈伟编写。朱战立对全书进行了认真和反复的修改。本书的最终出版得到了许多人的帮助,刘天时和韩家新老师就本教材的编写大纲等问题和作者进行过多次深入探讨,马刚老师和研究生杨锦锋、李红雷、杨萌、秦金祥等协助作者做了许多工作。在本书完成之际,一并向他们表示诚挚的感谢。

尽管作者在写作过程中投入了大量的时间和精力,但由于水平有限,错误和不足之处仍在所难免,敬请读者批评指正。

编著者

2004 年 9 月 1 日

目 录

第 1 章 概述	1
1.1 Java 语言简介	1
1.1.1 Java 语言的起源	1
1.1.2 Java 语言的版本	2
1.1.3 Java 语言的特点	2
1.2 Java 语言的运行机制	3
1.3 Java 开发运行环境 JDK	5
1.3.1 JDK 安装	5
1.3.2 JDK 下 Java 程序的编辑、编译与运行	7
1.4 两种 Java 程序	9
1.4.1 Java Application	9
1.4.2 Java Applet	12
习题 1	13
第 2 章 Java 语言基础	14
2.1 标识符	14
2.2 变量和常量	15
2.2.1 变量	15
2.2.2 常量	16
2.3 基本数据类型	16
2.4 赋值语句	18
2.5 运算符和表达式	19
2.5.1 运算符及其分类	19
2.5.2 运算符的优先级	24
2.5.3 表达式	24
2.6 流程控制语句	25
2.6.1 分支语句	26
2.6.2 循环语句	30
2.6.3 break 语句和 continue 语句	35
2.6.4 return 语句	36
2.7 程序注释	37
2.8 数组	37
2.8.1 一维数组	37
2.8.2 二维数组	42
2.8.3 不规则的二维数组	44
2.9 字符串	46

习题 2	47
第 3 章 类和对象	51
3.1 面向对象程序设计	51
3.1.1 面向对象程序设计的基本概念	51
3.1.2 类	52
3.1.3 对象	52
3.2 类	52
3.2.1 类声明	52
3.2.2 类主体设计	53
3.2.3 构造方法	55
3.3 对象	56
3.3.1 main 方法	56
3.3.2 对象的创建和初始化	57
3.3.3 对象的使用	59
3.3.4 垃圾对象的回收	59
3.3.5 实例成员变量与类成员变量	60
3.3.6 实例方法与类方法	61
3.3.7 方法的重写	62
3.4 包	63
3.4.1 包的建立方法	64
3.4.2 包的使用方法	65
3.4.3 包的访问权限	66
3.4.4 系统定义的包	69
3.5 内部类	70
3.6 类的封装性	71
3.7 设计举例	72
习题 3	75
第 4 章 类与继承	77
4.1 面向对象的基本概念: 继承	77
4.2 继承	78
4.2.1 子类和父类	78
4.2.2 创建子类	79
4.2.3 方法的三种继承形式	83
4.2.4 方法的多态性	86
4.3 抽象类和最终类	88
4.3.1 抽象类	88
4.3.2 最终类	90
4.4 接口	90
4.4.1 定义接口	90
4.4.2 实现接口	91

4.4.3 系统定义的接口	93
习题 4.....	94
第 5 章 Java API 基础	96
5.1 Java API 综述	96
5.2 语言包 (java.lang) 简介	97
5.2.1 Object 类	97
5.2.2 System 类	98
5.2.3 Class 类.....	100
5.2.4 Runtime 类	100
5.2.5 Float 类.....	101
5.2.6 String 类	101
5.2.7 Math 类.....	103
5.3 实用包 (java.util) 简介	104
5.3.1 Arrays 类	104
5.3.2 Vector 类	105
5.3.3 Data 类和 Calendar 类	105
5.3.4 Enumeration 接口.....	107
5.4 综合应用举例	107
习题 5.....	110
第 6 章 图形用户界面	111
6.1 图形用户界面和 AWT	111
6.1.1 图形用户界面的基本概念	111
6.1.2 AWT 简介	111
6.2 Component (组件) 类及其子类	112
6.2.1 Component (组件) 类	113
6.2.2 Container (容器) 类及其子类	114
6.2.3 Component 类的其他子类	114
6.3 MenuComponent (菜单组件) 类及其子类	121
6.4 AWT 中的绘图方法和常用类	123
6.4.1 Component (组件) 中的绘图方法	123
6.4.2 Color (颜色) 类	124
6.4.3 Font (字体) 类	125
6.4.4 Graphics (图形工具) 类	125
6.5 布局管理器	127
6.5.1 FlowLayout 类	128
6.5.2 BorderLayout 类	129
6.5.3 GridLayout 类	130
6.6 Java 的事件处理	131
6.6.1 事件处理的基本过程	131
6.6.2 Java 的事件处理	132

6.7 设计举例	139
习题 6	143
第 7 章 Java 小程序	145
7.1 Java 小程序概述	145
7.2 Java 小程序的特点、设计方法和运行环境	146
7.2.1 Java 小程序的特点	146
7.2.2 Java 小程序的基本设计方法和运行环境	146
7.3 Java 子程序的生命周期	148
7.3.1 Applet 类的继承关系	148
7.3.2 Java 子程序的生命周期	148
7.4 HTML 与 Applet	150
7.4.1 与 Applet 相关的 HTML 属性简介	150
7.4.2 HTML 文件和 Applet 的数据传递	151
7.5 两种典型的 Applet 程序设计	152
7.5.1 在 Applet 中加入图像	153
7.5.2 Applet 中的人机交互	154
7.6 Applet 的安全限制和 JAR 文件	156
7.6.1 Applet 的安全限制	156
7.6.2 JAR 文件	156
习题 7	157
第 8 章 异常处理	158
8.1 异常和异常处理的两种方法	158
8.1.1 异常的基本类型	158
8.1.2 if-else 形式的异常处理方法	158
8.1.3 Java 的异常处理方法	160
8.2 Java 的异常类	161
8.3 Java 的异常处理方法	163
8.4 异常的抛出和处理	167
8.4.1 在同一个方法中抛出异常和处理异常	168
8.4.2 抛出异常和处理异常的方法不是同一个方法	169
8.5 自定义的异常类	170
习题 8	171
第 9 章 输入/输出流	173
9.1 数据流的概念	173
9.1.1 输入流和输出流	173
9.1.2 字节流和字符流	174
9.1.3 Java 的标准数据流	174
9.2 基本输入/输出类	175
9.2.1 InputStream (字节输入流) 类	175
9.2.2 OutputStream (字节输出流) 类	178

9.2.3	Reader (字符输入流) 类	181
9.2.4	Writer (字符输出流) 类	183
9.2.5	FileReader 和 FileWriter (字符文件输入/输出流) 类	184
9.3	对象流	186
9.4	文件的操作	189
9.4.1	File (文件) 类及其应用	189
9.4.2	RandomAccessFile (随机存取文件) 类	193
习题 9		195
第 10 章	多线程	196
10.1	线程的基本概念	196
10.1.1	进程和线程	196
10.1.2	线程的生命周期和状态	197
10.2	Thread (线程) 类和 Runnable (可运行) 接口	197
10.2.1	Thread (线程) 类	197
10.2.2	Runnable (可运行) 接口	200
10.3	线程的状态和状态控制	202
10.3.1	线程的生命周期和状态	202
10.3.2	线程分组	203
10.3.3	线程的优先级	205
10.4	线程间的互斥	207
10.4.1	共享资源问题	207
10.4.2	互斥线程的设计方法	210
10.5	线程间的同步	212
10.5.1	共享资源的同步问题	212
10.5.2	同步线程的设计方法	215
10.6	综合应用举例	218
习题 10		221
第 11 章	数据库应用	223
11.1	JDBC 和 SQL 简介	223
11.1.1	SQL 简介	223
11.1.2	JDBC 简介	227
11.2	建立应用程序和数据库连接的环境配置	228
11.3	数据库应用编程	230
11.3.1	建立连接	230
11.3.2	操作数据库	233
11.3.3	处理操作结果	237
习题 11		241
第 12 章	网络通信	243
12.1	网络通信的基本概念	243
12.1.1	通信主体的定位	243

12.1.2	TCP 协议和 UDP 协议	245
12.1.3	网络程序设计的基本方式	245
12.2	使用 URL 类访问网络资源	246
12.2.1	资源定位器 URL 和 URL 类	246
12.2.2	URLConnection 类和 InetAddress 类	249
12.3	连接通信	252
12.3.1	Socket 和连接	252
12.3.2	Socket 类和 ServerSocket 类	252
12.3.3	Client/Server 结构的通信实例	254
12.4	数据报通信	259
习题 12		261
第 13 章	JSP 简介	262
13.1	网络服务和动态网站的基本概念	262
13.1.1	计算机网络和网络服务	262
13.1.2	网站和 JSP	263
13.2	JSP 的原理和特点	264
13.2.1	Servlet 和 JSP	264
13.2.2	JSP 的运行机制	264
13.2.3	JSP 的主要特点	265
13.3	运行环境的建立	265
13.3.1	安装支持 JSP 的服务器	265
13.3.2	JSP 运行环境的测试	266
13.4	JSP 的基本语法和内置对象	268
13.4.1	JSP 的基本语法	268
13.4.2	JSP 的指令语句	270
13.4.3	JSP 的内置对象	271
13.5	应用举例	274
习题 13		278
参考文献		279

第1章 概述



教学要点

本章内容主要包括 Java 语言的基本概念 (包括 Java 语言的起源、版本、特点等), Java 语言的运行机制, Java 语言的开发运行环境 JDK (包括 JDK 的安装和 JDK 的使用方法), 两种 Java 程序 (Java Application 和 Java Applet) 的基本形式。

要求了解 Java 语言的基本概念, 掌握 Java 语言的运行机制和开发运行环境 JDK, 了解两种 Java 程序的基本形式。

本章主要是让读者对 Java 语言和 Java 的开发运行环境有一个基本的了解; 内容主要包括 Java 语言的发展过程、版本变化、特点以及基本的运行机制, Java 语言的 JDK 开发环境, 以及用 JDK 和 JBuilder 开发 Java Application 和 Java Applet 两种类型程序的编辑、编译和运行过程。

1.1 Java 语言简介

1.1.1 Java 语言的起源

1991 年, Sun 公司为了向家用电子消费市场进军, 成立了一个代号为 Green 的项目组。其目标是开发一个分布式系统, 让人们可以利用网络远程控制家用电器。由于家用电器制造商众多、且制造标准各异, 所以项目组希望新系统应当具有独立于软件平台的特征, 并且新系统要安全易用。开始时, 项目组采用当时广泛使用的 C++ 语言进行系统开发。但是由于 C++ 语言太复杂, 安全性也难于满足要求, 所以不得不放弃, 转而研究设计出了一套新的程序设计语言用于系统的开发。这个新的程序设计语言就是 Java 语言的前身, 被命名为 Oak (橡树)。

Oak 以 C++ 语言为蓝本, 吸收了 C++ 中符合面向对象程序设计要求的部分, 同时加入了一些满足网络设计要求的部分。可惜的是, 由于一些商业上的原因, Sun 公司在以 Oak 为程序设计语言投标一个交互式电视项目时, 却未能中标, 这使得 Oak 语言的进一步发展一度遇到很大的问题。

20 世纪 90 年代中期, WWW 的影响在 Internet 上越来越大, WWW 浏览器开始在市场上出现。很多有远见的人意识到, 计算机网络和网络应用的浪潮即将到来。1994 年, Green 项目组成员在认真分析计算机网络应用的特点后, 认为 Oak 满足网络应用要求的平台独立性、系统可靠性和安全性等要求。为了展示 Oak 在网络应用方面的优良性能, 项目组用 Oak 设计了一个称为 WebRunner (后来称为 HotJava) 的 WWW 浏览器。1995 年 5 月 23 日, Sun 公司正式发布了 Java 和 HotJava 两项产品。

Java 语言一经推出,就受到了业界的关注。Netscape 是第一个认可 Java 语言的公司。1995 年 8 月,Netscape 公司将 Java 解释器集成到它的主打产品 Navigator 浏览器中。接着,Microsoft 公司在 Internet Explorer 浏览器中认可了 Java 语言,从而使 Java 终于站稳了脚跟,开始了自己的发展历程。

1.1.2 Java 语言的版本

到目前为止, Sun 公司先后发布了五个主要的 Java 语言版本。其发布时间和主要特点如下。

① Java 1.0: 1995 年推出的主要用于网页设计的版本,目前所有的浏览器都支持该版本。

② Java 1.1: 1997 年推出的一个版本,在用户界面和事件处理方面有改进,并且加入了 JavaBean 组件技术。

③ 带有 JDK 1.2 的 Java 2: 1998 年发布的具有重大改进的版本,在图形化用户界面、数据库互连以及其他许多方面做了改进。这个版本也称作带有 JDK 1.2 的 Java 1.2 版本。由于该版本具有重大改变,所以从 Java 1.2 版本后的 Java 语言也称作 Java 2 语言。

④ 带有 JDK 1.3 的 Java 2: 2000 年发布的版本,在多媒体应用、编译速度等方面做了改进。这个版本也称作带有 JDK 1.3 的 Java 1.3 版本。

⑤ 目前的最新版本是 JDK 1.4 版本。

JDK 是 Java Development Kit (软件开发工具集)的缩写。JDK 为用户提供了进行 Java 开发的、基于 DOS 环境的开发工具。Sun 公司新版本的帮助文档中也把 JDK 称作 Java SDK (Java Software Development Kit)。

Java 开发工具集可以在 Sun 公司网站 (<http://java.sun.com>) 上免费获取。这也是 Java 语言迅速发展的一个重要因素。当 Java 语言有较大改进时, Sun 公司总是同时推出新的 Java 语言版本和支持新版本的 Java 开发工具集版本。所以, Sun 公司的 JDK 新版本总是第一个支持 Java 新版本功能的开发工具,这通常比其他的 Java 开发工具早六个月到一年。

除了 Sun 公司的 JDK 外, Sun 公司和其他软件公司还推出了许多支持 Java 开发的工具。常见的 Java 开发工具还有:

① Sun 公司的 Forte for Java——这是一个基于 Windows 环境集成开发环境;

② Borland 公司的 JBuilder——这也是一个基于 Windows 环境的、集成开发环境,使用非常方便。

从 2000 年 12 月开始, Sun 公司又推出了 J2ME, J2SE, J2EE 三个版本。

① J2ME——Java 2 Micro Edition。嵌入式 Java 消费电子平台。J2ME 使嵌入式 Java 技术成为无处不在的计算模式。为新的企业、商务、娱乐和通信等应用领域提供解决方案。

② J2SE——Java 2 Standard Edition。用于工作站、PC 机的 Java 标准平台。

③ J2EE——Java 2 Enterprise Edition。可扩展的企业级应用 Java 2 平台。J2EE 是分布式企业软件组件架构的规范,具有灵活性、简化的集成性、便捷性,以及 J2EE 服务器之间的互操作性等特点。

1.1.3 Java 语言的特点

1996 年, Java 语言的主要设计者、Green 项目组的 James Gosling 和 Henry McGilton 在 Java 语言白皮书中,说明了他们最初的 Java 语言设计目标和实现这些目标的主要方法。

① 简单、面向对象和与现有高级语言形式类似。Java 语言是一种简单的、不用太多学

习训练就能掌握的语言，是一种面向对象的程序设计语言，是一种和目前流行的高级语言（如 C++ 语言）形式类似的语言。

② 鲁棒性和安全性。使用 Java 语言能生成高可靠性的软件，这主要靠 Java 语言的编译时检查和运行时检查两层检查机制实现。Java 语言主要用于分布式的网络环境，因此安全性是非常重要的。Java 语言限制应用程序从外部侵入系统传播病毒或破坏用户的文件系统。

③ 结构中立（architecture-neutral）。在网络环境下，用 Java 语言编写的应用程序必须适应各种各样的硬件结构和各种各样的操作系统平台，并能与其他程序设计语言接口协同工作。Java 语言实现这样的设计目标的主要方法是，先编译产生字节码（字节码是一种不依赖于任何硬件和操作系统的中间代码），然后用解释的方法产生最终在具体计算机上运行的机器码。

④ 高性能。通常，解释型程序设计语言运行速度较慢。Java 语言是一种半解释型语言，但 Java 语言程序的解释执行能全速进行，不用检查运行时的环境（因该环境已在编译时保证）。另外，Java 语言的自动垃圾回收线程定义成优先级很低的后台线程，该线程在系统空闲时启动执行，这也保证了 Java 语言的运行速度。

⑤ 解释型、多线程和动态性。只要安装了 Java 解释器，Java 字节码就能直接在任何计算机上解释执行。这种解释执行方式是简单的、递增式的，对系统要求很低。这能极大地方便用户程序的开发，并使开发过程很快。Java 的多线程机制可提高复杂应用程序的运行速度，并且这种多线程机制有系统提供的同步功能做安全保证，这些在网络应用环境下是十分必要的。Java 的对象绑定是动态的，类也是在被要求的时候才被从指定的任意位置导入的。这种动态性可保证不断地把最新的应用系统改进，方便地融合进应用系统中。这必将促进基于 Internet 的电子商务发展。

从上述 Java 语言设计者最初的设计目标，以及目前各种使用 Java 语言的应用系统使用情况来看，Java 语言具有如下显著特点。

① Java 是一种面向对象高级程序设计语言。面向对象技术是目前软件设计的主流和普遍使用的技术。和 C++ 等语言相比，Java 语言是一种纯面向对象的高级语言。

② 独立于各种操作平台。Java 采用先按编译方式翻译到字节码，然后再把字节码按解释方式翻译到具体机器的机器语言。这种半编译、半解释的高级语言翻译方式，使采用 Java 语言编写的应用系统具有独立于各种硬件环境和各种操作系统平台的特点，真正做到程序代码一次编写、多次使用。

③ 安全性好：安全性是网络应用系统必须考虑的重要问题，Java 语言从最初设计就充分考虑了安全性问题，杜绝了病毒传播和用户重要信息泄露的各种可预见的渠道。

④ 多线程：现在的操作系统都支持多任务和分时，允许在同一时间运行多个程序。同一时间运行多个程序主要有多进程和多线程两种方式，而多线程是一种效率更高的方式。Java 语言不仅支持多线程，而且为程序设计人员提供了实现线程同步机制的类库包。同步机制能保证多线程正确运行。

1.2 Java 语言的运行机制

目前大多数的高级程序设计语言，其应用程序编译产生的机器指令都与具体机器的硬件一定程度上相关。这种情况的后果是：当一个应用软件完成最后的编码设计后，在不同的硬件环境下编译生成的机器码是不一样的。因此，编译生成的可执行文件，只能在与编译时相

同的机器硬件环境下运行。否则，要么运行不成，要么运行时会出现错误。

Java 语言解决上述问题的方法是采用半编译、半解释的方式把高级语言的程序代码翻译成机器语言代码。为此，设计了一种称为字节码的代码。**字节码（Bytecode）**是一种和任何具体机器环境无关的中间代码。在 Java 程序编写完成后，首先，通过 Java 编译器把 Java 语言翻译成字节码，然后，通过 Java 解释器把字节码翻译成机器码。由于此种翻译方式既有编译又有解释，所以称作半编译、半解释方式。这样的过程如图 1.1 所示。

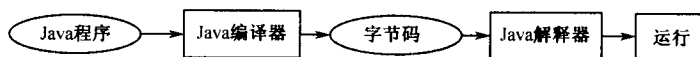


图 1.1 Java 语言程序的运行流程

Java 语言程序文件以 .java 为后缀。Java 程序编写完后，用开发环境下的编译器编译生成字节码，字节码文件以 .class 为后缀。这样的应用系统交付给用户后，只要用户计算机上安装有 Java 虚拟机，就可以把字节码翻译成相应计算机上的机器码。**Java 虚拟机（Java Virtual Machine，简称 JVM）**上有一个 Java 解释器，Java 解释器按解释方式把字节码翻译成具体硬件环境和操作系统平台下的机器码。这也就是说，Java 程序是相同的，只是不同运行平台上的 Java 虚拟机不同。图 1.2 说明了 Java 虚拟机的工作方式。

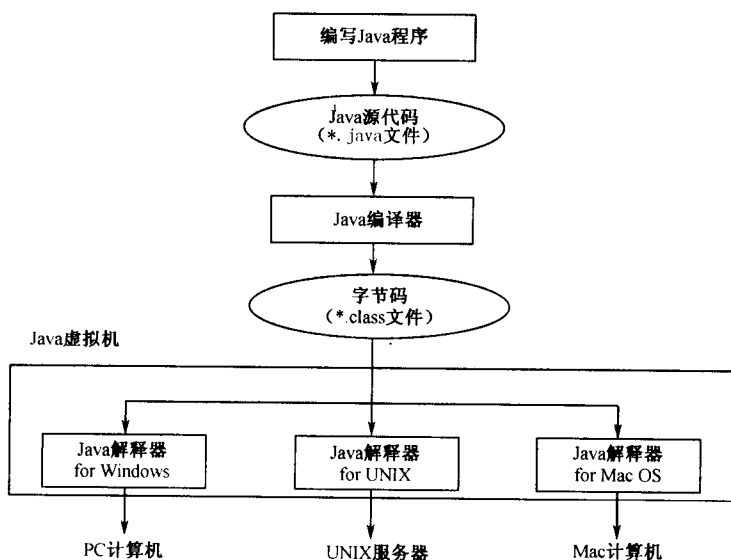


图 1.2 Java 虚拟机的工作方式

Java 语言这种一次编写、任意运行（write once, run anywhere）的方式有效地解决了目前大多数高级程序设计语言需要针对不同的机器编译产生不同的机器码的问题，即硬件环境和操作系统平台的异构问题。特别地，用 Java 语言开发的软件主要用在网络环境下。网络环境下应用软件的开发模式多是 B/S（Browse/Server）模式，即应用软件放在服务器端，客户端通过浏览器来进行应用。B/S 模式的应用系统特别需要异构环境的支持。

嵌入 HTML 中的 Java 程序称作 Applet 程序（小程序）。开发人员编写好 Applet 程序后，用开发环境下的编译器编译生成字节码文件，并把这样的字节码文件和 HTML 文件放在服务器上。当某个客户通过浏览器浏览相应服务器上的相应网页时，就把相应的 HTML 文件和 Applet 下载到了客户的计算机上，浏览器上都安装有 Java 虚拟机。此时，Java 虚拟机中

的 Java 解释器将按解释方式执行该 Applet，并把执行结果显示在网页上。

读者也许有疑问：既然只要在浏览器上安装了解释器就能解决应用软件运行环境的异构问题，那么为什么还需要先做一次编译，生成字节码呢？这是因为，如果把 Java 源程序在浏览器上按解释方式运行，那么客户计算机的运行速度将非常慢。

为了提高 B/S 模式客户计算机的响应速度，一方面，网络的频宽和流速在不断提高；另一方面，Java 虚拟机的性能也在不断提高。目前，用硬件直接实现的 Java 虚拟机正在加紧开发研究中。当 Java 虚拟机采用硬件直接实现时，其运行速度将大大快于目前的运行速度。

1.3 Java 开发运行环境 JDK

为开发和运行 Java 程序，计算机上必须安装相应的开发环境和开发工具。本节介绍 DOS 环境下的 Java 开发运行环境 JDK。Windows 环境下的 Java 开发运行环境 Jbuilder 在 1.4 节结合实例给出。

1.3.1 JDK 安装

1. 程序清单

为了建立基于 JDK 的 Java 程序开发运行环境，需准备以下一些程序和文档。

① j2sdk 1_3_1-win.exe (JDK 开发包) (不同的版本后面的序号不同，最新的是 j2 sdk 1_4-win.exe)。

② j2sdk1_3_1-doc.zip (相应的帮助文档)。帮助文档只是为了学习用，也可没有。

③ Uedit32i.exe 或记事本 (文本编辑器)。

前两个文件可以在 Sun 公司网站 (<http://java.sun.com/products>) 下载。Sun 公司在不断推出新的 JDK 版本，并把新版本在网站上及时发布。如果读者在该网站发现了新的 JDK 版本，也可下载新的 JDK 版本。

由于 JDK 没有提供一个 Windows 下的可视化的集成开发环境，所以必须自己准备编辑器。UltraEdit、记事本、EditPro 等都可以用来编写 Java 源程序。

2. 安装过程

(1) 安装 JDK 开发包

运行 JDK 开发包 j2sdk1_3_1-win.exe，安装过程中，可以设置安装路径并选择组件，也可按系统默认的安装路径和默认的安装组件安装。作者设置的安装路径为 C:\，因此，安装完成后，所有的 JDK 文件和子文件夹都在 C:\jdk1.3.1 文件夹中。

安装成功后，C:\jdk1.3.1 中的文件夹及用途如表 1.1 所示。

表 1.1 JDK 的文件夹结构

文 件 夹 名	用 途
bin	包含编译器、解释器等可执行文件
demo	包含源代码的程序示例
include	头文件，用于本地机的 C 语言
include-old	头文件，用于兼容支持旧接口
jre	Java 运行时环境的根路径
lib	可执行程序使用的包文件

为了能正确使用 JDK，需要手工配置一些环境变量，Windows 98 和 Windows NT/2000 的配置略有不同，下面分别介绍。

(2) Windows 98 的环境配置

在 C:\autoexec.bat 文件中，添加以下内容：

```
set path=%path%;c:\jdk1.3.1\bin
set classpath=.;c:\jdk1.3.1\lib
```

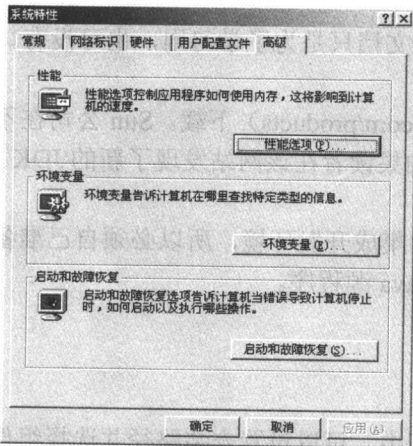
path 语句设置了 Java 编译运行程序的路径。其中，%path%表示保持系统原来的设置不变，C:\jdk1.3.1 为 Java 程序安装路径。set classpath 语句设置了 Java 包的路径，其中最前面的“.”表示在当前工作路径下可执行 Java 程序(若设置中无此圆点运行时会出现错误)，c:\jdk1.3.1\lib 为 Java 系统类包的路径。当然，如果安装的 Java 版本不同，或安装的路径不同，上述内容要做相应的改变。

上述修改完成后存盘，并重新启动计算机，这样就完成了环境设置。

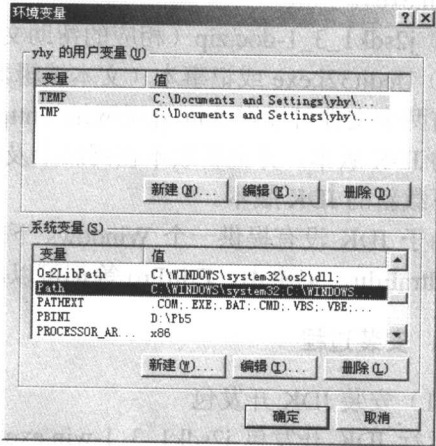
注意：上述 autoexec.bat 文件中不允许有空格，否则系统不能正常运行。

(3) Windows NT/2000 的环境配置

在桌面“我的电脑”图标上单击鼠标右键，选择“属性”选项，出现“系统特性”对话框，如图 1.3 (a) 所示，在“高级”选项卡中单击“环境变量”按钮。在“系统变量”栏框中，找到 Path 选项。其操作窗口如图 1.3 (b) 所示。



(a)



(b)

图 1.3 “系统特性”与“环境变量”对话框

单击“编辑”按钮，将“;c:\jdk1.3.1\bin”添加到变量值文本框中，单击“确定”按钮结束编辑变量。其操作窗口如图 1.4 所示。

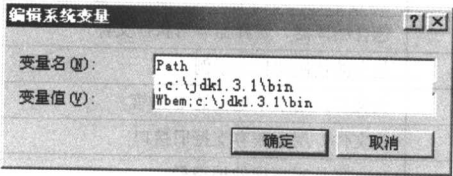


图 1.4 “编辑系统变量”对话框

同样方法再设置 `classpath` 环境变量，变量值要添加的字符串为 `“.;c:\jdk1.3.1\lib”`。

设置完成后，在 DOS 窗口下，输入 `java` 和 `javac` 后，如果出现其用法的参数提示信息，则安装正确；否则安装不正确，需重新安装。

(4) 安装 JDK 帮助文档

JDK 帮助文档可帮助读者学习 JDK 的使用方法和完整地掌握 JDK 的各种资料。要安装 JDK 帮助文档，用户的下载软件中必须包括 `j2sdk1_3_1-doc.zip` 文件。这是一个压缩文件，其安装方法是直接运行该压缩文件，此时将在 `C:\jdk1.3.1` 路径下再创建一个 `jdk1.3` 文件夹。用浏览器或 Windows 的资源管理器浏览 `index.html` 文件，可看到帮助文档主页如图 1.5 所示。该窗口左上窗显示 Java 的包，左下窗显示所选包相应的类，右窗显示相应类中成员变量和方法的说明文档。

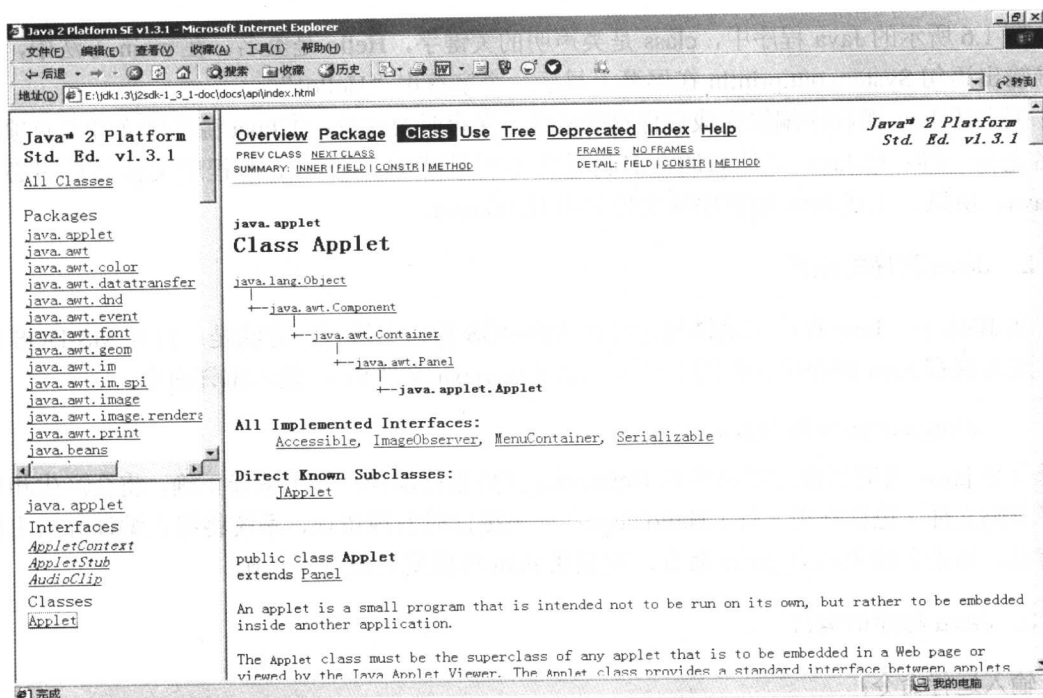


图 1.5 JDK 帮助文档主页

(5) 建立一个工作文件夹

工作文件夹用于放置读者自己编写的 Java 程序。如果读者使用 JDK 工具开发 Java 程序，考虑到 JDK 工具是在 DOS 环境下的命令操作方式，建议读者创建的工作文件夹名使用英文字母，不要使用汉字。

1.3.2 JDK 下 Java 程序的编辑、编译与运行

为了清楚地解释 JDK 下 Java 程序的编辑、编译与运行过程，本节用一个最简单的例子说明该过程。

【例 1.1】 设计显示 `Hello!` 字符串的 Java 程序，并用 JDK 运行该程序。