



21 世纪高职高专信息技术教材

Java 程序 设计教程

北京希望电子出版社	总策划
袁启昌	主 编
王趾成	副主编
袁海宁	编 著
张光瑞	



清华大学出版社

Java 程序 设计教程

清华大学出版社

清华大学出版社



21 世纪高职高专信息技术教材

Java 程序 设计教程

北京希望电子出版社 总策划
袁启昌 主 编
王趾成 副主编
袁海宁 张光瑞 编 著

 科学出版社
www.sciencep.com

内 容 简 介

本书是“21 世纪高职高专信息技术教材”中的一本,本书力求使读者能够快速、轻松地学会用 Java 编程。本书紧扣 Java 程序设计语言的特点,从面向对象的基本概念和 Java 语言特性的多个方面逐一展开,由浅入深、循序渐进地介绍了 Java 语言的基础知识和主要特征。

本书在注重系统性和科学性的同时,力求突出其实用性;在介绍相关编程原理和基础知识的前提下,着重利用丰富实用的例子来展示 Java 编程技术的真正魅力。

本书内容丰富,结构合理,注重实用性,既可用作高职高专、成人高等教育计算机及相关专业的教材,也可作为广大计算机爱好者自学 Java 语言时的参考书。

需要本书或需要得到技术支持的读者,请与北京中关村 083 信箱(邮编:100080)发行部联系,电话:010-82702660, 010-82702658, 010-62978181 转 103 或 238, 传真:010-82702698, E-mail: tbd@bhp.com.cn

图书在版编目(CIP)数据

Java 程序设计教程/袁启昌主编. —北京:科学出版社,
2005.3

21 世纪高职高专信息技术教材

ISBN 7-03-014651-4

I. J... II. 袁... III. Java 语言—程序设计—教材

IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 122604 号

责任编辑:王玉玲

/ 责任校对:佳 宜

责任印刷:媛 明

/ 封面设计:梁运丽

· 科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

北京市媛明印刷厂印刷

科学出版社发行 各地新华书店经销

*

2005 年 3 月第 一 版 开本:787×1092 1/16

2005 年 3 月第一次印刷 印张:14 1/2

印数:1—5 000 册 字数:334 000

定价:23.00 元

21世纪高职高专信息技术教材编委会名单

(排名不分先后)

主 任 高 林

副主任 谢玉声

袁启昌

胡伏湘

陆卫民

委 员

阮东波

龙 超

杨丽群

王东红

罗映峰

侯晓华

连晋平

冯矢勇

杨章静

唐伟奇

徐 萍

尹 静

慕东周

李 淼

田 更

朱作付

李超燕

吴 军

杨旭东

景鹏森

杨金龙

米 昶

陈孟建

崔俊杰

孙 杰

宗小翀

陈翠娥

唐燕青

韦 伟

陈 春

刘 毅

袁海宁

徐建华

邱建国

曹冬梅

郑明红

蒋建强

陈彦许

韩素华

王趾成

崔会军

张光瑞

总 序

高等职业教育目前已成为我国高等教育的重要组成部分,对于推动我国社会主义现代化建设起着不可忽视的作用。计算机教育在整个高职教育中有着举足轻重的地位,因为计算机的普及已经涉及到各个行业。对于传统的学习计算机知识的方法即理论为主、应用为辅的教学模式,相对高职教育来说有些不太适合,针对这种情况,就需要一些符合高职教育特点的教材来满足这种需求。

为解决教材供需不平衡的矛盾,北京希望电子出版社与全国高等学校计算机基础教育研究会高职高专专业委员会联合组织国内十几所高职院校,聘请“双师”型教师共同编写针对高职特点的教材 30 多种,以及实训类教材 10 多种,并请专家论证了本套教材的体系、风格、结构、内容等方面的可行性与可操作性。该系列教材体现“重在能力素质培养”的目标,结合教育部的教学大纲要求,在实用性、新颖性、可读性几个方面都有所突破。

高职教材建设是教学改革重要的环节,高等职业技术教育专业设置要与劳动力市场需求相结合,教学内容与国家职业标准相衔接。采取“订单教学”的校企合作培养模式,实行学业文凭和职业资格两种证书制度,使一线技术人才培养实现教学与市场“零距离”、毕业生上岗“零适应期”。这种以市场为导向实行的订单教学,能够直接为用人单位培养实用型人才,是一条富有特色的职教之路,可以保证同学们将来在就业和升学两条渠道上有最大的发展空间。所以,高校就要突出应用技能培养的办学特色,按照人才市场供求信号进行学科、专业和教学内容的调整,以适应社会需要。在培养学生的知识、能力、技能方面都要与其他综合性本科院校有所区别。

本系列教材就是遵循这种订单式教学的需要,一方面是设定系统理论知识的教材,这种教材的内容按照“必需、够用”的原则,构筑坚实的具有高职特色的理论体系基础;另一方面是训练职业动手能力的实训教材,按照“切实、实用”的原则,培养动手能力强的人才。以上两种教材相互配合,既可以单独使用,也可以配套使用。

高职教材建设还在探索中,如何能满足企业对人才的需求,跟上时代发展的步伐,这些都是亟需解决的问题。本丛书旨在抛砖引玉,希望更多的优秀教师参与到教材建设中来,真诚希望广大教师、学生与读者朋友在使用本丛书过程中提出宝贵意见和建议,为下一次的修订与改版做准备,使本丛书日臻完美。

若有投稿或建议,请发至本丛书出版者电子邮件: textbook@bhp.com.cn。

前 言

以网络技术为基础的知识经济正不断地改变着社会文化生活的各个方面，Internet 已成为 21 世纪社会发展最重要的基石之一。由美国 SUN 公司开发的新一代面向对象的程序设计语言 Java，以其独有的、与网络紧密结合的特点，已经成为 Internet 领域功能最强大、最有前途的编程语言之一。

本书紧扣 Java 编程语言的特点，从 Java 的平台独立性、面向对象、容器、多线程、异常处理等多个方面逐一展开，从不同的方面展现了 Java “编写一次、随处运行”的精髓。

本书的叙述浅显易懂，循序渐进，即使是没有编程经验的新手，通过本书的学习也可以较快地掌握 Java 编程技术，并将本书介绍的大量实例应用到实际开发中。本书中没有复杂难懂的专业术语，所有的专业术语将结合大部分读者可以理解的实际例子进行解释。在介绍最新的知识及技术动向时，用大量的程序实例加以讲解，便于读者理解和巩固所学的知识。

本书共分 11 章，第 1 章概要介绍了 Java 语言的一些基本概念、Java 语言的特点并简述了 Java 程序的结构和开发过程；第 2 章介绍了 Java 最基础的编程知识，通过这章的学习可以掌握最基本的编程概念，为今后的编程打下基础；第 3 章通过一些实例介绍了 Java 的数组和字符串；第 4 章是本书的重点，包括面向对象的基本概念，并详细介绍了 Java 的面向对象程序开发方法；第 5 章介绍了 Java 编程中很重要的容器类的使用方法；第 6 章介绍了 Java 的异常处理机制，以使本书读者能掌握异常处理的正确概念，写出更加安全和稳定的 Java 程序；第 7 章介绍了在 Java 中如何处理流、文件以及如何控制输入/输出；第 8 章介绍了 Java 如何实现与平台无关的图形用户界面；第 9 章介绍了 Java 的 Applet 编程，包含了许多有趣的实例；第 10 章介绍了如何在 Java 中实现多线程；第 11 章介绍了数据库的基本概念以及如何使用 JDBC 编程技术访问数据库。

本书可作为高职高专、成人高等教育计算机及相关专业 Java 课程的教材，也可作为对 Java 语言和面向对象编程技术感兴趣的一般读者的自学用书。

本书由袁启昌教授主编，王趾成副主编，袁海宁、张光瑞、张恒杰、崔俊杰等编著。虽然本书的作者在写作的过程中尽了最大努力，但由于时间和水平有限，书中难免有疏漏和不妥之处，敬请广大读者不吝赐教。

编 者

目 录

第1章 Java 语言概述.....1	2.4.7 表达式与运算符优先级.....24
1.1 Java 的发展和应用前景.....1	2.5 流程控制语句.....25
1.1.1 Java 的产生和发展.....1	2.5.1 分支语句.....25
1.1.2 Java 的应用前景.....2	2.5.2 循环语句.....31
1.2 Java 的特点.....2	2.6 习题.....35
1.3 Java 运行环境的建立.....4	第3章 数组和字符串.....37
1.3.1 Java SDK 简介.....4	3.1 一维数组.....37
1.3.2 Java SDK 运行环境配置.....5	3.1.1 一维数组的定义.....37
1.3.3 Java SDK 工具.....8	3.1.2 一维数组元素的引用.....38
1.3.4 Java 程序的编译.....8	3.1.3 一维数组的初始化.....38
1.3.5 Java 程序的执行.....9	3.1.4 一维数组程序举例.....39
1.4 开始第一个 Java 程序.....10	3.2 多维数组.....40
1.5 Java 的编程风格.....11	3.2.1 二维数组的定义.....40
1.6 习题.....11	3.2.2 二维数组元素的引用.....41
第2章 Java 语言基础.....12	3.2.3 二维数组的初始化.....41
2.1 Java 语言的基本组成.....12	3.2.4 二维数组举例.....41
2.1.1 标识符.....12	3.3 字符串.....42
2.1.2 关键字.....13	3.3.1 String 类.....43
2.1.3 注释.....13	3.3.2 创建 String 对象.....43
2.2 数据类型.....13	3.3.3 String 类的常用函数.....44
2.2.1 常量与变量.....14	3.3.4 StringBuffer 类.....47
2.2.2 整型数据.....14	3.4 习题.....49
2.2.3 浮点型（实型）数据.....15	第4章 Java 面向对象程序设计.....50
2.2.4 字符型数据.....16	4.1 面向对象的程序设计方法.....50
2.2.5 布尔型数据.....17	4.1.1 类、对象和消息.....50
2.2.6 举例.....17	4.1.2 面向对象程序主要特性.....52
2.3 各类数值型数据间的混合运算.....18	4.2 Java 的对象特性.....53
2.3.1 自动类型转换.....18	4.2.1 Reference（引用，参考）.....53
2.3.2 强制类型转换.....19	4.2.2 建立对象.....54
2.4 运算符和表达式.....19	4.3 Java 的类特性.....56
2.4.1 算术运算符.....19	4.3.1 Java 的类结构.....56
2.4.2 关系运算符.....21	4.3.2 函数（方法）、参数表和返回值.....57
2.4.3 布尔逻辑运算符.....22	4.3.3 static 数据成员和 static 函数.....58
2.4.4 位运算符.....23	4.4 函数重载.....60
2.4.5 赋值运算符 =.....23	4.5 对象的初始化.....61
2.4.6 条件运算符.....24	

4.5.1	成员定义初始化.....	61
4.5.2	用构造函数初始化.....	63
4.5.3	static 数据成员的初始化.....	67
4.6	关键字 this	70
4.7	package (包) 的使用	72
4.8	Java 的访问权限.....	76
4.8.1	Java 的访问权限饰词.....	76
4.8.2	package 访问权限.....	76
4.8.3	public 访问权限.....	79
4.8.4	private 访问权限.....	80
4.8.5	protected 访问权限.....	83
4.8.6	Class 访问权限和默认 package	84
4.9	Java 的组合和继承.....	86
4.9.1	组合的使用.....	86
4.9.2	继承的使用.....	90
4.9.3	类的多态性.....	96
4.10	抽象类和接口.....	100
4.10.1	Java 的抽象类.....	100
4.10.2	Java 的接口.....	102
4.11	习题.....	104
第 5 章	Java 的容器.....	106
5.1	Java 的容器简介.....	106
5.2	Collection 族系介绍.....	108
5.2.1	Collection 接口介绍.....	108
5.2.2	List 容器介绍.....	110
5.2.3	Set 容器介绍.....	113
5.3	Map 族系介绍.....	117
5.3.1	Map 简介.....	117
5.3.2	Map 容器介绍.....	117
5.4	迭代器.....	121
5.4.1	Iterator 迭代器.....	121
5.4.2	ListIterator 迭代器.....	123
5.5	习题.....	126
第 6 章	Java 的异常处理机制.....	127
6.1	异常处理简介.....	127
6.1.1	异常处理的概念.....	127
6.1.2	异常处理的结构.....	127
6.1.3	异常的种类.....	128
6.2	使用 Java 的异常处理.....	130

6.2.1	try...catch...finally 语句.....	130
6.2.2	throws 和 throw 关键字.....	137
6.3	创建属于自己的异常类.....	139
6.4	习题.....	143
第 7 章	流、文件及其输入/输出.....	144
7.1	流和文件.....	144
7.1.1	流的基础知识.....	144
7.1.2	文件.....	145
7.2	基本的输入/输出类.....	145
7.2.1	InputStream 类.....	145
7.2.2	OutputStream 类.....	146
7.2.3	PrintStream 类.....	147
7.2.4	文件的输入/输出.....	147
7.2.5	File 类.....	149
7.3	习题.....	154
第 8 章	图形用户界面 GUI.....	155
8.1	常用组件.....	155
8.1.1	图形用户界面的概念.....	155
8.1.2	建立和添加组件.....	156
8.1.3	组件 JFrame 和 JLabel.....	158
8.1.4	组件 JTextField 和 JButton.....	159
8.1.5	组件 JCheckBox 和 JComboBox.....	161
8.1.6	组件 JSlider 和菜单.....	165
8.2	布局管理器.....	169
8.2.1	布局管理器 FlowLayout.....	169
8.2.2	布局管理器 BorderLayout.....	171
8.3	Java 的事件处理.....	174
8.3.1	事件处理 4 要素.....	174
8.3.2	事件处理的 3 种做法.....	175
8.4	习题.....	176
第 9 章	Applet 程序设计.....	177
9.1	Applet 概述.....	177
9.1.1	第一个 Java Applet.....	177
9.1.2	在 HTML 文件中嵌入 Applet... ..	179
9.2	Applet 的设计基础.....	180
9.2.1	Applet 的类层次.....	180
9.2.2	Applet 程序生命周期与实例分析.....	181

9.3	Applet 应用例子	184
9.3.1	加载图片的 Applet	184
9.3.2	在 Applet 中使用超级链接	185
9.3.3	播放音乐的 Applet	188
9.3.4	传递参数给 Applet 程序	190
9.4	Applet 的安全限制	194
9.5	习题	194
第 10 章	Java 多线程	195
10.1	线程	195
10.1.1	线程体	195
10.1.2	线程的状态	196
10.2	线程体的构造	198
10.2.1	通过继承构造线程体	198
10.2.2	通过接口构造线程体	200
10.3	生产者-消费者问题	201
10.4	习题	206

第 11 章	Java 与数据库	207
11.1	数据库简介	207
11.2	结构化查询语言简介	208
11.3	Java 的数据库访问方法: JDBC	210
11.3.1	ODBC 到 JDBC 的发展历程	210
11.3.2	JDBC 技术概述	211
11.4	JDBC 接口介绍	213
11.4.1	JDBC 数据库访问概述	213
11.4.2	JDBC 驱动设置	215
11.4.3	JDBC Statement 语句设置	216
11.4.4	JDBC 结果集设置	217
11.5	一个简单 JDBC 实例	218
11.6	习题	219
附录 A	命名规范与技巧	221
附录 B	推荐读物	223

本章重点内容:

- Java 的发展和应用前景
- Java 的特点
- Java 运行环境的建立
- 开始第一个 Java 程序
- Java 的编程风格

第 1 章

Java 语言概述

欢迎进入 Java 的世界! Java 语言是一种简单、面向对象、分布式、安全、结构中立、可移植、直译、高性能、多线程的程序语言,在 21 世纪的今天,Java 语言是网络程序设计人员的首选。本章将对 Java 的发展进行回顾并展望其发展前景,同时介绍了 Java 语言的特点,并介绍了如何构建一个 Java 开发环境,本章最后还将引入一个非常简单的 Java 入门程序。通过本章的学习,读者可以了解 Java 的发展史,并掌握如何建立一个 Java 开发环境,为本书今后章节的学习打下基础。

1.1 Java 的发展和应用前景

1.1.1 Java 的产生和发展

如今非常热门的 Java 语言,最初是源自于 Sun 公司(Sun Microsystems)内部的绿色计划,其主要目的是开发能用来在电视、烤面包箱等“消费性电子产品”上进行交互式操作的软件,Sun 公司最初构思是用 C 或 C++来开发这类软件,但很快他们发现 C 或 C++并不适合,因为用 C 或 C++必须针对控制“消费性电子产品”的软件运行的不同平台来编写不同的软件,而且由于 C 或 C++本身的复杂性,不容易编写可靠性高的程序,这样将使程序的稳定性降低,因此绿色计划开发小组决定放弃 C 或 C++作为软件的开发语言,而决定重新开发一种新的语言,于是在 1990 年设计出一个名为“Oak”的新语言。

“Oak”就是 Java 的前身,是一种简单、可靠并且平台独立的程序语言,后来因为在申请商标时,发现这个名称已经被另一个程序语言所使用了。由于很多程序员喜欢喝一种叫做 Java 的热咖啡,根据这样的灵感,Java 这个名称从众多名称中脱颖而出,这也就是 Java 图标中总是出现一杯热咖啡的由来。

Java 的出现一开始并没有引起人们的注意,直到 1994 年下半年,随着 Internet 的迅猛发展,环球信息网 WWW 的快速增长,Java 的开发小组发现 Java 这种跨平台的语言非常适合应用到网络上,于是他们开发了第一个支持 Java Applet 的浏览器——Hot Java。Java 在 WWW 上的优异表现使得它逐渐成为 Internet 上受欢迎的开发与编程语言。Java 的“Write Once, Run Anywhere”的特性使得用 Java 开发的软件可以不用修改或重新编译而直接应用于任何计算机上。同时 Java 的开发人员基于网络对 Java 进行了一系列的改进,融合了 C 和 C++等语言的优点,形成了现在这套与众不同的面向对象的通用程序设计语言。

Java (JDK1.0) 正式发表于 1995 年 5 月,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 80 年代以来计算机界的一件大事”,微软总裁

比尔·盖茨在悄悄地观察了一段时间后，不无感慨地说：“Java 是 长时间以来最卓越的程序设计语言”，并确定微软整个软件开发的战略从 PC 单机时代向着以网络为中心的计算机时代转移，而购买 Java 则是他重大战略决策的实施部署。

2004 年 1 月美国“勇气号”火星探测器登陆火星，控制火星车的软件使用的语言正是 Java，这使 Java 的足迹拓展到了宇宙。

因此，Java 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了新的挑战。

1.1.2 Java 的应用前景

Java 语言有着广泛的应用前景，大体上可以从以下几个方面来考虑其应用：

- 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 Web 页面的动态设计、管理和交互操作设计等；
- Intranet（企业内部网）上的软件开发（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；
- 其他应用类型的程序。

Java 由原本专为“消费性电子产品”所设计的程序语言，演变成为改变网络世界的新一代程序语言，这是大家始料不及的。Java 将会不断地扩大它的应用领域，以网络为发展基础，把设计领域延伸至无线通讯、电子商务、航空航天等相关领域。

1.2 Java 的特点

Java 是一种简单、面向对象、分布式、强韧性、安全、结构中立、可移植、直译、高性能、多线程而且动态的语言（出自 Sun 公司的 Java 语言白皮书）。对这些特性的具体解释如下（有些特性读者可以在读完全书以后重新回味，这样可以加深对 Java 特性的理解）。

1. 简单性

Java 简单特性的定义就是让用户易学易懂。C/C++ 语言历史悠久，为了让原来使用 C/C++ 的程序员能够很快的接受 Java，很容易上手来开发程序，Java 引入了 C/C++ 的很多语法概念，同时 Java 也删除了在 C/C++ 内使用较少或是容易被混淆的功能，如 Java 删除了 C++ 的“指针”。不可否认，C++ 的指针功能强大，但强大的背后是复杂性，而且使用指针极可能会产生内存存取错误。所以有人说 Java 语言是 C++——（即移去累赘、困难部分的 C++）。

Java 还提供了一个很棒的功能就是内存垃圾自动回收机制的功能，程序员可以专注于程序设计本身，而不必担心内存泄漏等 C/C++ 常见错误，在内存管理上不必分心。这样也大大简化了程序设计的复杂性。

2. 面向对象

Java 语言是一种面向对象（OO，Object Oriented）的程序设计语言，无论是 Java 应用程序，还是 Java 小程序，它们都是以类为基础构建的。类是 Java 语言的基础语法元素。面向对象的众多特性在 Java 的语法结构里面均有完美的实现。因此我们应该以面向对象的方式来分析、设计、编码 Java 的应用程序。

有关面向对象的程序设计方法理论本书第 5 章将为读者介绍，读者也可以参考一些经典的面向对象设计书籍（如《设计模式》等）以提高自己的面向对象分析设计能力。

3. 分布式

Java 的在互联网的成功也使 Java 成为分布式程序设计的首选语言。在设计 Java 语言时开发小组就已经考虑到网络传输协议，所以 Java 能够轻松处理网络协议。这样 Java 程序所使用的对象不必局限在本地机器上，当 Java 的对象分布在其他主机上的时候，Java 可以利用网络来打开或是存取远程对象，就像存取本机的对象一样方便。

4. 强韧性

许多有 C/C++ 经验的读者一定会对 C/C++ 的指针的复杂型有深刻印象，对 C/C++ 指针的误用经常会造成系统的崩溃，Java 删除了 C/C++ 的指针功能，降低了语言的复杂型并避免了由于用户的失误而造成系统不稳定的问题。

此外，除了删除指针功能外，Java 还提供了内存垃圾自动回收机制和异常处理机制，来排除程序运行时可能出现的错误。内存垃圾自动回收机制可以自动回收程序不用的内存空间，避免内存不足。而异常处理机制更是内嵌在 Java 语言的语法结构里，异常处理机制将程序所有的异常情况（如除数为 0）集合在一起，并在语法层面上提供支持，使程序设计更加健壮，程序流程更加合理。

5. 安全性

网络语言必须考虑安全性，因此 Java 的设计充分考虑了安全的因素。例如，由于屏蔽了指针，Java 不允许用户直接对内存进行存取操作，这样互联网上的黑客就无法通过在程序中读取或是修改内存地址中的内容来达到危害用户安全的目的。

6. 结构中立性

Java 的主要特色在于它的平台无关性，所谓平台无关性是指 Java 程序不受限于计算机硬件设备和操作系统种类，只要计算机中有 Java 的运行环境（JVM，Java Virtual Machine，Java 虚拟机），Java 的程序就可以在那台计算机上运行。这也正是 Sun 公司强调的 Java 优势：Write Once，Run Anywhere 一次编写，到处运行。

7. 可移植性

Java 的结构中立性确保了 Java 程序的可移植性，在 Java 中严格规定了数据类型和类结构的标准，这样就可以使各种数据结构在不同的计算机平台上都适用。除此之外，Java 的

链接库还定义了一些可移植的接口以及这些类在不同的计算机平台上的定义结构，这些都使 Java 具有良好的可移植性。

8. 直译性

Java 是一种直译式的语言，这说明，Java 经过编译后并不直接生成机器码（C++则相反），而是被编译成字节码（Byte Code），然后通过 Java 的直译器 JVM 将字节码转换成机器码来运行。JVM 是 Java 定义用来运行字节码的环境，也就是说 Java 编译后的字节码并不直接在计算机平台上运行，而是在 JVM 上运行。不同的计算机平台上（如 Windows, Unix, Macintosh 等）的 JVM 有不同的规格，这样就可以使 Java 的程序不受限于计算机平台，只要某种计算机平台具有其上的 JVM，Java 程序就可以在这种平台上运行，这也是 Java 程序平台无关性的运行原理。

9. 高性能

Java 字节码是一种中间码，虽然不能像 C/C++那样通过编译产生运行文件（如：.exe）来直接运行，而必须加载 JVM 才能运行，但 Java 字节码在编译时就考虑到机器码的产生，这使字节码在转换成机器码时的速度和性能逼近 C/C++的运行文件。

10. 多线程

在生活中多线程的例子很多很多，比如我们可以一边上网，一边听音乐。在程序设计中，线程的概念简单说就是程序中的一个连续的控制流程。多线程是指程序中包含多条执行路径。在一个程序中可以同时运行多个不同的线程来执行不同的任务，即允许单个程序创建多个并行执行的线程来完成各自的任務。Java 具有开发多线程程序的功能，可以有效地利用处理器资源。

11. 动态性

Java 运行时需要的对象（Object）是根据程序运行情况被动态加载的，具体来说，Java 程序并不是在程序一开始运行时将所需对象全部加载，而是在程序运行的过程中——动态加载，当 JVM 发现当前所需的对象尚未被加载时，便先在本机上查找，如果不存在则通过某种机制在网络查找并加载，这样可以有效利用内存资源。

1.3 Java 运行环境的建立

学习 Java 语言，一定要亲自动手编写代码。Java 有许多开发环境，如 Sun 公司的 Java SDK, Borland 公司的 JBuilder, Microsoft 公司的 Visual J++等。Sun 公司的 Java SDK 是免费工具，可以到 Sun 公司网站或其他提供软件下载的网站去下载。本书使用的开发工具采用的是 Java SDK。

1.3.1 Java SDK 简介

Java SDK（Java Software Development Kit）即 Java 软件开发工具包，目前提供下载的 SDK 标准版软件最新正式版本是 1.4.1，有不同的操作系统的不同版本，请读者根据自己的

操作系统选择相应版本下载。

下载并安装 Java SDK 开发工具

可以从 <http://java.sun.com> 下载最新的 SDK 开发工具, 笔者使用 Windows 操作系统, 所以下载后的软件名称是 j2sdk-1_4_1-win.exe。读者也可以根据需要在 Java API 说明文档 j2sdk-1_4_1-doc.zip。下载完成后运行 j2sdk-1_4_1-win.exe 即可进行开发工具的安装, 安装时要指定安装分区和目录。本书选用安装到默认路径 C:\jdk1.4。



提示

本书用 <JAVA-HOME> 表示 SDK 安装的目录, 比如读者将 SDK 装在 C:\jdk1.4, 则 <JAVA-HOME> 代表的路径即为 C:\jdk1.4。

1.3.2 Java SDK 运行环境配置

Java SDK 的运行环境配置主要是两个方面, 即 Path 和 ClassPath 的设置。在 Window 98、Windows 2000 中的 Path 和 ClassPath 的设置不同, 如果你没有正确设置 Path 和 ClassPath, 那么就不能运行和测试 Java 程序。

1. Path 的设置方法

(1) Windows 98 操作系统: 在 C 盘的根目录下找到 Autoexec.bat 文件, 用文本编辑工具打开, 找到包括 “set Path=” 字符这行代码, 在代码后面加入 <JAVA-HOME>\bin, 如果你的 Java SDK 安装路径是 C:\jdk1.4, Autoexec.bat 中的 “set Path=” 字符将是

```
set path=c:\windows;c:\windows\command;c:\jdk1.4\bin
```

目录之间的间隔使用 “;”, 如果 Autoexec.bat 文件里没有 “set Path=” 这行字符, 则需添加这行字符。

(2) Windows 2000 和 Windows NT 操作系统:

1) 打开系统特性中的 “高级” 对话框。

在 Windows 2000 中用鼠标指向 “我的电脑”, 单击鼠标右键, 在弹出的快捷菜单中选择 “属性” 命令, 打开系统特性对话框, 单击 “高级” 标签, 如图 1-1 所示。

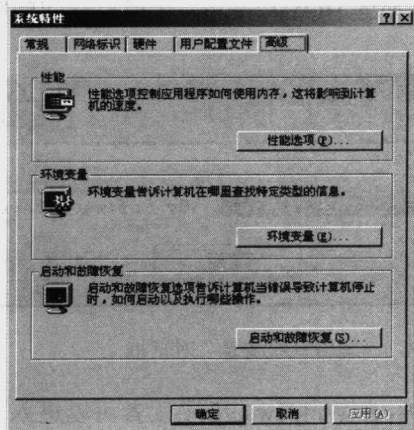


图 1-1 系统特性 “高级” 对话框

2) 在“高级”标签内单击“环境变量”按钮，或者按 Alt+E 键，显示如图 1-2 所示的窗口。

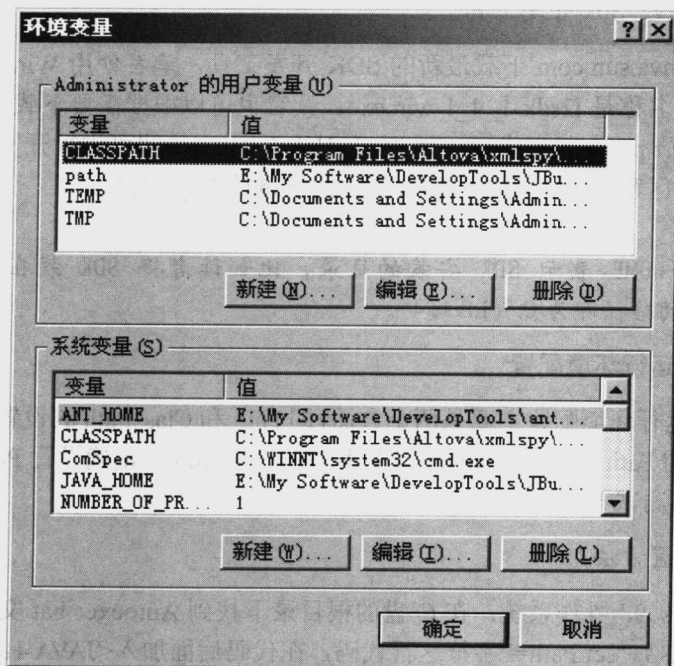


图 1-2 环境变量对话框

该窗口有两项内容，Administrator 的用户变量和系统变量，它们的具体区别是作用范围的不同，Administrator 是针对管理员用户所设置的变量，当你用其他用户登录时，该变量将不存在，系统变量是针对所有用户设置的变量，一旦设置该变量后，任何用户登录 Windows 2000 系统都共享相同的路径设置。

3) 在系统变量中找到 Path 变量。

单击“编辑”按钮，如果没有 Path 变量，单击“新建”按钮添加 Path 变量，在弹出的编辑系统变量里进行编辑，加入<JAVA-HOME>\bin，注意目录间间隔用“;”，单击“确定”按钮，如图 1-3 所示。



图 1-3 “编辑系统变量”对话框

4) 测试 Path 设置。

为检测 Path 设置是否正确，在 Windows 98 中读者可以进入 DOS 方式，在 Windows 2000 中读者可以单击“开始”按钮，单击“运行”，输入“cmd”，进入命令行状态。在命

令提示符下输入“javac”，回车，如果出现如图 1-4 所示的提示则表示 Path 设置正确。

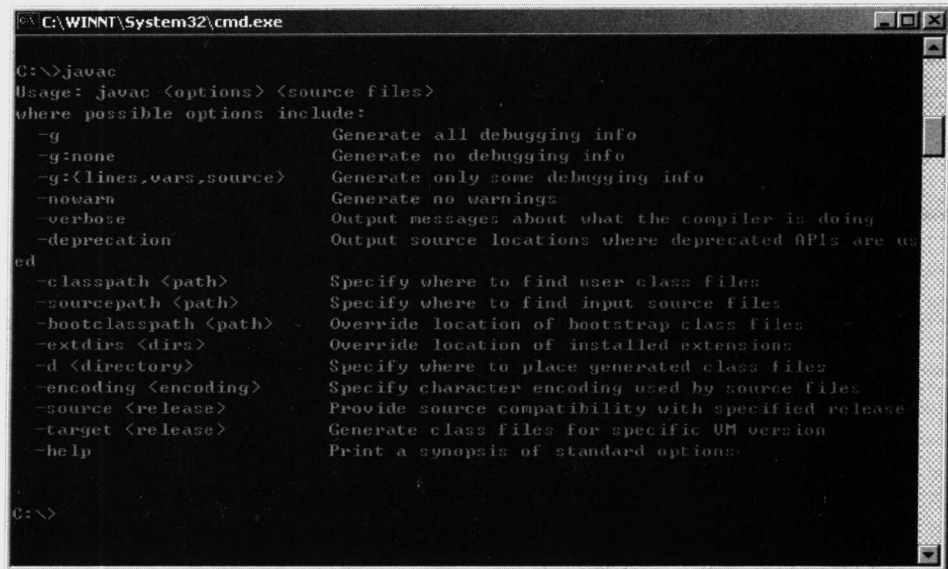


图 1-4 Path 设置正确后显示的界面

2. ClassPath 的设置

设置 ClassPath 的目的是让 Java 虚拟机 JVM 找到所需要的类库，JVM 寻找类库的顺序如下：

A. 启动类库 这些类库组成了 Java 平台，包括 rt.jar 和 i18n.jar 两个文件。这个启动类库会在 JVM 运行时自动加载。

B. 扩展类库 这些类库保存在<JAVA-HOME>\jre\lib\ext 目录下，这个类库也是 JVM 运行时自动加载的。

C. 用户自定义类库 我们需要设置的正是这个类库，因为这个类库 JVM 不会自动加载，需要用户设置路径。

不同操作系统下用户自定义类库：

(1) Windows 98 操作系统下的设置。

在 C 盘的根目录下找到 Autoexec.bat 文件，用文本编辑工具打开，找到包括“set Classpath=”字符这行代码，在代码后面加入自定义类库的路径，以及当前路径，当前路径表示方法是“.”，缺少当前路径也会使程序运行出现错误。例如，你所使用的类库路径在“d:\MyProgram\lib”则“set Classpath=”字符将是

```
set Classpath= d:\MyProgram\lib;.
```



注意

这行字符是以“.”结束，表示将当前路径加入 Classpath。目录之间的间隔使用“;”，如果 Autoexec.bat 文件里没有“set Classpath=”这行字符，则需添加这行字符。

(2) Windows 2000 和 Windows NT 操作系统下的设置。