



新编21世纪高职高专计算机系列规划教材

Java 语言程序设计

北京希望电子出版社 总策划
张群哲 主 编
周 翼 翁建红 副主编

科学出版社
www.sciencep.com



新编21世纪高职高专计算机系列规划教材

Java 语言程序设计

北京希望电子出版社 总策划
张群哲 主 编
周 翼 翁建红 副主编

 科学出版社
www.sciencep.com

内 容 简 介

Java 是当今最流行也是最有前途的面向对象程序设计语言。本书共分 10 章, 将 Java 语言基础和面向对象程序设计方法结合起来, 以大量实例由浅入深地讲述了 Java 语言基础、面向对象编程方法、Java 流处理、GUI 编程、多线程编程、网络编程等 Java 实用技术。

全书内容丰富、讲练结合, 每章都配有相应的练习和实验指导, 并在最后一章设置了综合编程指导, 使读者通过本书的学习, 能学到真正实用的技术, 从而解决实际问题。

本书既可以作为高职高专、成人高等教育院校计算机及相关专业的教材, 也可作为广大爱好者自学 Java 的参考书。

需要本书或技术支持的读者, 请与北京清河 6 号信箱 (邮编: 100085) 发行部联系, 电话: 010-82702660 010-82702658 010-62978181 转 103, 传真: 010-82702698, E-mail: tbd@bhp.com.cn。

图书在版编目 (CIP) 数据

Java 语言程序设计 / 张群哲主编. —北京: 科学出版社,
2005.8

新编 21 世纪高职高专计算机系列规划教材

ISBN 7-03-016045-2

I. J... II. 张... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 085542 号

责任编辑: 王超辉 / 责任校对: 孙 红
责任印刷: 媛 明 / 封面设计: 刘孝琼

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 8 月第 一 版 开本: 787×1092 1/16
2005 年 8 月第一次印刷 印张: 14 3/8
印数: 1-3000 册 字数: 327 000

定价: 20.00 元

新编 21 世纪高职高专计算机系列规划教材编委会

- 主 任： 陈火旺 全国工科院校计算机专业教学指导委员会主任
中国工程院院士
- 副主任： 沈复兴 全国高等师范学校计算机教育研究会副理事长
北京师范大学信息科学学院院长
- 何炎祥 武汉大学计算机学院院长
- 桂卫华 中南大学信息科学与工程学院院长
- 李仁发 湖南大学计算机与通信学院院长
- 唐汝元 湖南张家界航空职业技术学院副院长
- 刘小芹 湖北武汉职业技术学院副院长
- 陆卫民 中国科学出版集团北京希望电子出版社社长

委 员： （按姓氏笔画为序）

于小川	牛军涛	王 立	王 虹	王亚林	冯玉东	石 磊
刘其群	刘晓魁	向长喜	曲宏山	朱乃立	朱志伯	朱国军
齐英兰	张凌雯	张惠敏	李 晨	李节阳	李国安	杨成卫
狄文辉	肖 衍	苏 玉	连卫民	陆立松	陈利军	陈桂生
周承华	易著梁	罗立红	郑明红	胡远萍	骆 刚	徐刚强
秦国防	崔清民	黄贻彬	黄振中	彭 勇	曾凡秩	蒋本立

秘 书： 李节阳

总 序

一本好书，是人生前进的阶梯；一套好教材，就是教学成功的保证。为满足培养应用型人才的需要，我们成立了本编委会。在明确高职高专应用型人才培养模式、培养目标、教学内容和课程体系的框架下，我们组织编写了本套规划教材。

为了使本套教材能够达成目标，编委会做了大量的前期调研工作，在广泛了解各高职高专学校的教学现状、学生水平、培养目标的情况下，认真探讨了课程设置，研究了课程体系。为了编写出符合教学需求的好教材，我们除了聘请一批计算机知名专家、教授作为本套教材的主审和编委外，还组织了一批具备较高的学术水平、丰富的教学经验、较强的工程实践能力的学术带头人和骨干教师来承担具体编写工作，从而编写出特色鲜明、适用性强的教材，以真正满足目前高职高专学校应用型人才培养的需要。教材编写采用整体规划、分步实施、在实践中检验提高的方式，分期分批地启动编写计划。编写大纲以及教材编写方式的确定均经过编委会多次认真讨论，以确保该套教材的高质量和实用性。

本套规划教材的主要特点是：

(1) 以服务教学为最高宗旨，认真做好教学内容的取舍、教学方法的选取、教学成果的检验工作。本套教材在教学过程中的有益反馈，都将及时体现在后续版本。

(2) 面向应用型高职高专，在保证学科体系完整的基础上把握好理论的深度和难度。注重理论知识与实践相结合，使学生通过实践深化对理论的理解，学会并掌握理论方法的实际运用。从而较好地培养学生的专业技能和实施工程的实用技术能力。

(3) 教材在内容编排上，力求由浅入深，循序渐进；举一反三，突出重点；语言简练，通俗易懂。采用模块化结构，兼顾不同层次的需求，在具体授课时可根据具体教学计划适当取舍内容。

(4) 教材采用“任务驱动”的编写方式，以实际问题引出相关原理和概念，在讲述实例的过程中将本章的知识点融入，通过分析归纳，介绍解决工程实际问题的思想和方法，同时，引入案例教学和启发式教学方法，便于激发学习兴趣。

(5) 在教材中加大实训部分的比重，使学生能比较熟练地应用计算机知识和技术解决实际问题，既注重培养学生分析问题的能力，也注重培养学生解决问题的能力。

(6) 大部分教材配有电子教案，从而更好地服务教学。

为编写本套教材，作者们付出了艰辛的劳动，编委会的各位专家进行了悉心的指导和认真的审定。书中参考、借鉴了国内外同类的优秀教材和专著，在此一并表示感谢。

我们衷心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书过程中提出宝贵意见和建议。

若有投稿或建议，请发电子邮件到 textbook@bhp.com.cn。谢谢！

新编 21 世纪高职高专计算机系列规划教材编委会

前 言

一、关于Java

Java 作为在 Internet 上最流行的编程语言,其发展非常迅速。作为一种优秀的面向对象的程序设计语言,Java 从 1995 年诞生以来,就以其具有无关性、安全机制、高可靠性和内嵌的网络支持等特点,成为当前网络应用程序编写的首选工具之一。

二、本书结构

本书主要包括 Java 应用程序编程及设计领域的许多内容,可以将本书作为第一门开发语言来学习,获得开发程序的初步经验。对于其他有高级语言编程经验的读者,则可以从本书中领略 Java 语言面向对象、易学易用的特点。

本书共分为 10 章,各章内容安排如下:

第 1 章 Java 概述。主要介绍程序设计语言的分类、面向对象程序设计的特点、Java 编程环境的设置和简单的 Java 程序编辑执行等内容。

第 2 章 Java 语言基础。主要介绍了 Java 的基本数据类型、数组的概念、运算符及其优先级、简单输入输出等内容。

第 3 章 流程控制。主要介绍了 Java 的语法基础、流程控制语句。

第 4 章 方法。主要介绍了方法定义与使用、参数传递、方法重载以及异常处理。

第 5 章 面向对象程序设计。主要介绍了面向对象程序设计的概念、类与继承、接口和包等内容。

第 6 章 输入输出系统。主要介绍了文件系统和流系统等内容。

第 7 章 图形用户界面和 Applet。主要介绍了基本的 AWT 组件、Swing 组件和事件处理、Applet 的创建与执行等内容。

第 8 章 多线程处理。主要介绍了线程的基本知识和使用方法、线程的通信与同步、用线程实现动画等内容。

第 9 章 网络编程技术。主要介绍了 Socket 的基础知识、单线程及多线程服务器和客户端程序、UDP 通信等内容。

第 10 章 综合应用编程。主要介绍了网络聊天室系统和简单媒体播放系统的设计方案与程序代码。

三、本书的特点

本书在编写过程中始终贯彻面向对象的方法，将 Java 基础知识与网络编程、多媒体技术等有机地结合起来，力图详细地介绍面向对象的概念及原则机制等。

本书内容丰富、结构合理，并结合大量实例和直观的图示来介绍 Java 程序设计的原理和方法。为帮助读者巩固所学的知识，每章都配有练习题和实验指导。

四、本书的适用对象

本书既可以作为高职高专、成人高等院校计算机及相关专业的教材，也可作为广大编程爱好者自学 Java 的参考书。

全书由张群哲任主编，周翼、翁建红任副主。全书由张群哲统稿。参加编写工作的人员有宋剑杰、邹宇、周志化、张群哲、周翼、翁建红等。在本书的编写过程中，参阅了大量文献资料和网上资料，在此一并表示衷心的感谢！

由于编者水平有限，书中错漏之处在所难免，敬请广大读者批评指正。

编 者

目 录

第1章 Java 概述.....1	本章实验 数据类型与运算符36
1.1 程序设计语言的分类.....1	第3章 流程控制38
1.1.1 按发展过程分类.....1	3.1 程序的逻辑控制流程38
1.1.2 按执行方式分类.....2	3.2 选择控制结构39
1.1.3 按思维模式分类.....2	3.2.1 由 if 语句引导的选择结构39
1.2 Java 简介3	3.2.2 由 switch 语句引导的选择结构42
1.2.1 Java 的历史和主要特点.....3	3.3 循环控制结构45
1.2.2 面向对象的几个基本概念.....5	3.3.1 for 语句45
1.2.3 Java 的运行及运行环境.....6	3.3.2 while 语句47
1.2.4 常见的 Java 集成开发环境.....11	3.3.3 do...while 语句48
习题 1.....11	3.3.4 循环嵌套48
本章实验 Java 开发环境12	3.3.5 三种循环结构的比较.....49
第2章 Java 语言基础.....14	3.3.6 break 语句50
2.1 Java 程序的基本构成要素14	3.3.7 continue 语句52
2.1.1 标识符.....14	习题 353
2.1.2 保留字.....14	本章实验 Java 流程控制54
2.1.3 分隔符.....15	第4章 方法56
2.2 数据类型.....15	4.1 方法的概念及作用56
2.2.1 常量与变量.....16	4.2 方法的定义与调用58
2.2.2 基本数据类型.....16	4.2.1 方法的定义格式.....58
2.2.3 数组.....19	4.2.2 方法的调用.....60
2.3 运算符.....23	4.3 参数传递和返回值60
2.3.1 算术运算符.....24	4.3.1 参数传递.....61
2.3.2 关系运算符.....25	4.3.2 方法的返回值.....63
2.3.3 逻辑运算符.....25	4.4 方法中变量的作用域64
2.3.4 位运算符.....27	4.5 方法的嵌套和递归调用66
2.3.5 赋值运算符.....27	4.5.1 方法的嵌套.....66
2.3.6 条件运算符.....28	4.5.2 递归调用67
2.4 表达式.....28	4.6 方法的重载69
2.4.1 类型转换.....29	4.7 异常处理71
2.4.2 优先级.....30	4.7.1 异常的概念.....71
2.4.3 常用的数值计算方法.....31	4.7.2 异常处理方式.....72
2.5 简单输入输出.....32	4.7.3 异常控制机制.....72
2.5.1 输出.....32	4.7.4 Java 定义的异常.....73
2.5.2 输入.....33	习题 478
习题 2.....35	

本章实验 Java 中方法的使用以及异常处理 ..78	6.5.1 Java 的 File (文件) 类118
第 5 章 面向对象程序设计.....80	6.5.2 File 类对象的重要方法.....119
5.1 类 (class)80	习题 6121
5.1.1 类定义.....80	本章实验 输入输出程序设计121
5.1.2 类对象.....81	第 7 章 图形用户界面和 Applet123
5.1.3 构造方法.....83	7.1 Java 的图形界面123
5.1.4 类和成员的修饰符.....85	7.1.1 AWT 与 Swing.....123
5.2 类的继承 (inheritance)87	7.1.2 图形界面元素.....124
5.2.1 继承的概念.....87	7.2 Swing 组件及其之间的层次关系125
5.2.2 类继承的实现.....88	7.2.1 一个例子.....126
5.2.3 抽象类和抽象方法.....91	7.2.2 容器.....128
5.3 接口 (interface)92	7.2.3 布局管理.....131
5.3.1 接口的定义.....93	7.3 事件机制135
5.3.2 接口的实现.....93	7.3.1 事件处理概述.....135
5.3.3 接口的继承.....94	7.3.2 常用事件处理.....135
5.4 包 (package)95	7.3.3 自定义事件.....141
5.4.1 创建包.....95	7.4 Applet 与 HTML144
5.4.2 使用包.....95	7.4.1 HTML 简介144
5.4.3 类及类成员的访问权限.....97	7.4.2 Applet 的生命周期.....146
5.4.4 Java 的应用程序接口 (API)98	7.4.3 Applet 的其他重要方法.....149
习题 5.....98	习题 7151
本章实验 类与对象.....99	本章实验 图形用户界面与 Applet.....152
第 6 章 输入输出系统.....101	第 8 章 多线程处理154
6.1 输入输出流的概念.....101	8.1 多线程概述154
6.1.1 概述.....101	8.1.1 多线程的概念.....154
6.1.2 java.io 包提供高级输入输出流101	8.1.2 Java 的线程模型.....155
6.2 输入输出类.....102	8.2 线程的基本结构和使用方法157
6.2.1 字节流 InputStream 和 OutputStream 类102	8.2.1 线程的生命周期.....157
6.2.2 字符流 Reader 和 Writer 类104	8.2.2 线程类的方法.....158
6.2.3 标准输入/输出.....105	8.2.3 多线程的实现.....159
6.3 文件的顺序访问.....106	8.2.4 线程的控制.....162
6.3.1 输入输出流操作的一般步骤.....106	8.3 线程的管理163
6.3.2 使用字节流访问文件.....107	8.3.1 线程同步.....163
6.3.3 使用字符流访问文件.....111	8.3.2 优先级.....167
6.4 文件的随机访问.....116	8.4 用于制作动画的线程169
6.4.1 建立随机访问文件流对象.....116	8.4.1 动画程序框架.....169
6.4.2 读写随机访问文件方法.....116	8.4.2 帧的画法.....171
6.4.3 文件指针及相关方法.....117	8.4.3 消除闪烁感.....172
6.5 文件管理.....118	8.4.4 使用图片.....173
	习题 8176

本章实验 Java 线程和多媒体程序设计	176
第 9 章 网络编程技术	178
9.1 网络基础.....	178
9.1.1 客户机/服务器体系.....	178
9.1.2 TCP/IP 协议.....	179
9.1.3 Socket 模式.....	181
9.1.4 IP 地址和端口	181
9.2 TCP Socket 应用	182
9.2.1 TCP 客户端.....	182
9.2.2 TCP 服务器.....	184
9.2.3 多线程服务器.....	185
9.2.4 Socket 实现用户登录认证.....	186
9.3 UDP Socket 应用	191
9.3.1 UDP Socket 基础	191
9.3.2 DatagramPacket 类	191
9.3.3 InetAddress 类.....	191
9.3.4 DatagramSocket 类	192
9.3.5 接收与发送数据报.....	192
9.3.6 UDP 通信实例.....	193
习题 9.....	195
本章实验 网络编程技术.....	196
第 10 章 综合应用编程	198

10.1 建立 C/S 模式聊天室服务器	
和客户端	198
10.1.1 实例目的.....	198
10.1.2 实现方法.....	198
10.1.3 程序代码.....	199
10.1.4 完善程序功能的建议.....	211
10.2 浏览图像和播放声音的 Java	
应用程序	211
10.2.1 实例目的.....	211
10.2.2 实现方法.....	212
10.2.3 程序代码.....	212
附录 习题参考答案	216
第 1 章.....	216
第 2 章.....	216
第 3 章.....	217
第 4 章.....	219
第 5 章.....	221
第 6 章.....	224
第 7 章.....	226
第 8 章.....	232
第 9 章.....	235

第1章 Java概述

教学目标

本章主要针对学习程序设计语言之前要了解的一些概念进行介绍，继而引导大家对面向对象和 Java 有一个初步认识，同时介绍 Java 的开发和运行环境。通过本章的学习，应当达到：

- 了解面向过程与面向对象
- 理解面向对象中的几个基本概念
- 学会建立Java的运行环境

1.1 程序设计语言的分类

程序设计语言是学习计算机技术的基础，它经历了较长的发展过程，也有许多不同的分类方法，下面将介绍几种对学习 Java 比较有帮助的分类。

1.1.1 按发展过程分类

按计算机系统发展的历程来看，程序设计语言大致可以分以下 4 种。

1. 机器语言

机器语言是二进制代码形式的机器指令集合，不同的机器有不同的机器语言，存储安排也由语言本身控制。以这种语言编制的程序运行效率极高，但程序很不直观，编写很简单的功能就需要大量代码，重用性差，而且编写起来效率比较低，很容易出现错误。

2. 汇编语言

汇编语言比机器语言直观，它将机器指令进行了符号化，并增加了一些功能，如宏、符号地址等，存储空间安排由机器完成，编程工作相对机器语言有了极大的简化，使用起来方便了很多，错误也相对减少。但不同的指令集的机器也有不同的汇编语言，程序重用性也很低。

3. 高级语言

高级语言是与机器语言不相关的一类程序设计语言，读写起来更接近人类的自然语言。因此，用高级语言开发的程序可移植性和重用性也要好得多。常见的高级语言有 Pascal、C 和 Basic 等，现在应用程序的设计多数都是使用高级语言。Java 就是高级语言的一种。

4. 第 4 代语言

一种还未成熟的语言。它具有一定的智能，更接近于日常语言，它对语言的概括更为抽象，从而使语言也更为简洁。

1.1.2 按执行方式分类

按照程序的执行方式的不同，程序设计语言可分成两大类。

1. 编译执行的语言

编译执行是在编写完程序后，通过特定的工具软件将源代码经过目标代码转换成机器代码，即可执行程序，然后直接交给操作系统执行，也就是说程序是作为一个整体来运行的。这类程序语言的优点是执行速度比较快，还有编译连接之后可以独立在操作系统上运行，不需要其他应用程序的支持；缺点是不利于调试，每次修改之后都要执行编译连接等步骤，才能看到其执行结果。当然，有些集成开发环境之间存在一定的依赖性，不同操作系统需要的编译器可能不相同，因此，在一个系统上编译的程序到另外一个系统上并不一定能够运行。常见的编译执行的程序语言有 Pascal 和 C 等。

2. 解释执行的语言

解释执行是程序读入一句执行一句，而不需要整体编译连接，这样的语言与操作系统的相关性相对较小，但运行效率低，而且需要一定的软件环境来做源代码的解释器。当然，有些解释执行的程序并不是使用源代码来执行的，而是需要预先编译成一种解释器能够识别的格式，再解释执行。例如本书中介绍的 Java 就是这样的一种语言。不过，解释执行的语言相对来说调试比较方便。常见的解释执行的程序语言有 Basic 和 Java 等。

1.1.3 按思维模式分类

程序设计语言总是需要以某种思维方式进行设计和实现，因此不同的语言可能有不同的思维方式。目前存在两种思维方式。

1. 面向过程的程序设计语言

所谓面向过程就是以要解决的问题为思考的出发点和核心，并使用计算机逻辑描述需要解决的问题和解决的办法。

针对这两个核心目标，面向过程的程序设计语言注重高质量的数据结构和算法，研究采用什么样的数据结构来描述问题，以及采用什么样的算法来高效地解决问题。在 20 世纪 70 年代和 80 年代，大多数流行的高级语言都是面向过程的程序设计语言，如 Basic、FORTRAN、Pascal 和 C 等。

这类语言面向求解问题的过程，而不依赖于计算机硬件，可移植性相对较好，在计算机所要解决的问题还不是非常复杂、使用的范围还不是非常广泛的条件下，是非常有效的解决问题的方法。但面向过程的程序设计语言有一个致命的缺点，它极度面向过程，即使需要解决的问题发生微小的变化，也会对程序本身产生很大的影响，也就是说需要程序员对程序做较大的改动。而且，不同的问题需要不同的程序解决，问题与解决程序几乎是一一对一的，以往的成果很难直接利用。因此，其可维护性和可重用性都比较差。随着计算机应用范围的扩大，面向过程的程序设计语言的缺点就更加明显地暴露出来。而解决这些问题的最有效方法就是另外一种面向对象的思维模式。

2. 面向对象的程序设计语言

面向对象 (Object-Oriented) 不仅仅是一种程序设计语言的概念，应该说是一种全新的

思维方式。简单的说,面向对象的基本思想就是以一种更接近人类一般思维的方式去看待世界,把世界上的任何一个个体都看成是一个对象,每个对象都有自己的特点,并以自己的方式做事,不同对象之间存在着通讯和交往,以此构成世界的运转。用计算机专业的术语来说,对象的特点就是他们的属性,而能做的事就是他们的方法。

面向对象方法大大提高了程序的重用性,而且从相当程度上降低了程序的复杂度,使得计算机程序设计能够对付越来越复杂的应用需求。常见的面向对象的程序设计语言包括 C++ 和 Java 等,其中最为突出的是 Java 语言。如今,在电子商务时代,以 Java2 企业版(J2EE)为主的模型更是成为一种事实上的电子商务平台的服务标准。因此,学习和掌握 Java 语言和面向对象技术在电子商务时代将大有用武之地。

1.2 Java 简介

在所有的面向对象程序设计语言当中,Java 是最纯粹,结构最清晰的一种语言,它严格遵守着面向对象的绝大多数思想和理念,是学习面向对象思想最好、最有效的一种语言。同时,Java 又具备构造非常复杂的应用的能力,现在以及未来的很多应用都将基于 Java 来开发,它的发展前景是不可估量的。

1.2.1 Java 的历史和主要特点

1. Java 语言的历史

任何事物的产生既有必然的原因也有偶然的因素,Java 语言的出现也验证了这一点。1991 年,美国 Sun Microsystems 公司(简称 Sun 公司)的某个研究小组为了能够在消费电子产品上开发应用程序,积极寻找合适的编程语言。消费电子产品种类繁多,包括 PDA、机顶盒、手机等等,即使是同一类消费电子产品所采用的处理芯片和操作系统也不相同,也存在着跨平台的问题。

Sun 公司的研究人员曾考虑是否可以采用 C++ 语言来编写消费电子产品的应用程序,但是研究表明,对于消费电子产品而言,C++ 语言过于复杂和庞大,并不适用,安全性也并不令人满意。于是,Bill Joy 先生领导的研究小组就着手设计和开发出一种语言,称之为 Oak。该语言采用了许多 C 语言的语法,提高了安全性,并且是面向对象的语言,但是 Oak 语言在商业上并未获得成功。

时间转到了 1995 年,互联网在世界上蓬勃发展,Sun 公司发现 Oak 语言所具有的跨平台、面向对象、安全性高等特点,非常符合互联网的需要,于是改进了该语言的设计,造就了一代成功的编程语言,Sun 公司将之取名为 Java 语言。

2. Java2 的版本

Java 语言根据其应用领域的不同分为不同的版本,由于 to 和 two 发音一样,所以把“Java to”简写为 Java2。Java2 主要有以下几个版本:

(1) J2ME (Java2 Micro Edition)

针对嵌入式技术。

(2) J2SE (Java2 Standard Edition)

针对桌面计算机开发（主要有网页中内嵌的小程序 Java Applet 和应用程序 Java Application）。

(3) J2EE (Java2 Enterprise Edition)

针对企业级的应用解决方案（主要有 JDBC、CORBA、EJB、JavaServlets、JSP 以及对 XML 的支持）。

各版本之间的关系如图 1-1 所示。

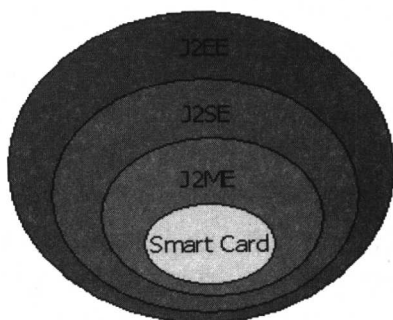


图 1-1 Java2 各版本之间的关系

3. Java 的特点

Java 之所以能够在计算机和网络技术高速发展的今天占据这样关键性的地位，主要是由以下特点决定的。

(1) 简单性

Java 语言是一种面向对象的语言，它通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。它的基本解释器及类的支持只有 40KB 左右，加上标准类库和线程的支持也只有 215KB 左右。

(2) 面向对象

Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏；而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法，实现了代码的复用。

(3) 分布性

Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他对象。

(4) 强壮性

Java 在编译和运行程序时，都要对可能出现的问题进行检查，以消除错误的产生。它提供自动垃圾收集来进行内存管理，防止程序员在管理内存时容易产生的错误，还提供了异常处理机制以防止系统的崩溃。另外，Java 在编译时还可捕获类型声明中的许多常见错误，防止动态运行时不匹配问题的出现。

(5) 安全性

用于网络、分布环境下的 Java 必须要防止病毒的入侵。Java 不支持指针，一切对内存

的访问都必须通过对象的实例变量来实现，这样就防止程序员使用“特洛伊木马”等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

(6) 体系结构中立

Java 解释器生成与体系结构无关的字节码指令，只要安装了 Java 运行系统，Java 程序就可在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示，Java 解释器得到字节码后，对它进行转换，使之能够在不同的平台上运行。

(7) 可移植性

与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同机器。同时，Java 的类库中也实现了与不同平台的接口，使这些类库可以移植。

(8) 解释执行

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时信息，使得连接过程更加简单。

(9) 高性能

和其他解释执行的语言如 Basic 不同，Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码，从而拥有较高的性能。

(10) 多线程和动态性

多线程机制使应用程序能够并行执行，而且同步机制保证了对共享数据的正确操作。

1.2.2 面向对象的几个基本概念

面向对象的概念不是一两句话就能够解释清楚的，读者的学习也是一个反复理解，逐步深入的过程，但无论如何总要有一个开始。在这里，我们只简单提几个基本概念，为引入 Java 程序做一个铺垫。

1. 类与对象

所谓对象 (Object) 就是真实世界中的实体，对象与实体是一一对应的，也就是说实现世界中每一个实体都是一个对象，它是一种具体的概念。而类 (Class) 是具备某些共同特征的实体的集合，它是一种抽象 (Abstract) 的概念，用程序设计的语言来说，类是一种抽象的数据类型，它是对具有相同特征实体的抽象。

2. 抽象

抽象就是把事物的共同点抽取出来，以同样的方式进行概要描述的一种过程。这种过程是提高程序重用性的根本原因，正是因为对许多类似的事物进行了抽象，过去已经产生的成果才可以在相同或类似的环境下重用。

3. 属性与方法

不同对象具有相同的特点，就可能抽象为一定的类，那么这些特点基本上可以分为两类：一类是描述对象静态状态的，就是对象的属性 (Attribute)；另一类是描述对象的动作，就是对象的方法 (Method)。属性和方法是一个对象所具备的两大基本要素，也是今后编程工作的核心。

4. 封装

属性和方法存在的意义是什么呢？无论一个对象方法的目的是什么，最终表现为对对象属性的操作，包括把对象的状态告知外界以及改变对象的状态。只要有足够的方法，就没有必要直接去操作对象属性，只要调用这些方法就可以实现要完成的任务，这种现象称为封装，它通过对象方法对其属性的操作把对象属性封装在一个对象内部。

同时，对于外界来说，这些对象方法到底是怎么工作的，我们都不需要去关心，只要知道它能实现的功能以及如何去调用这种功能就可以了，例如开车的操作将改变汽车的状态，但一个司机并不需要了解这些，他只要会使用“开车”这个方法就行了。这使得程序的设计和实现有效的分开，使得方法之间的依赖性大大降低，对象之间的独立程度也大为提高，从而有效地提升了程序的可维护性。

5. 继承

一个类可以从它的父类继承状态和行为。继承为组织和构造软件程序提供了一个强大的和自然的机理。总的说来，对象是以类的形式来定义的。每一个子类从父类中继承了状态。同样，每一个子类继承父类的方法。

1.2.3 Java 的运行及运行环境

Java 有两类程序，即 Java 应用程序 (Java Application) 和 Java 小程序 (Java Applet)，前者是在命令中运行的独立的应用程序，而后者需要嵌入 HTML (Hyper Text Mark Language, 超文本标记语言) 在浏览器中执行。本节将分别介绍这两类程序的编译执行方法。

1. 编写并执行 Java 程序的步骤

编写并执行 Java 程序的步骤可以分为 3 个阶段 (以使用 JCreator Pro 为例)。

JCreator Pro 是一种常用的 Java 编辑软件，它具有强大的行、列编辑功能，是一般的编辑器所无法比拟的，而且，在 JCreator Pro 中可以使用自定义的配置文件，从而在编写 Java 程序时，系统自动把关键字、常量、变量等不同元素区分开，有助于程序员减少语法错误。这个软件可以从网上下载，使用起来非常简单，这里不作具体介绍。

在选择编辑器时，大家千万不要使用像 Word 这类带有格式的文本编辑工具，因为它会在看不到的情况下，给文本加入很多格式信息，这些信息是 Java 解释器所不能识别的。

(1) 编写源代码

在这个阶段，我们直接在 JCreator Pro 中输入源代码。程序编写好以后就可以保存到软盘或硬盘的某一目录下，这个文件称为源程序，储存时，其后缀名必须是 .java。建议读者把所有源文件都保存到一个目录下，便于运行。

(2) 编译源代码

虽然 Java 是解释型语言，但它仍有一个类似编译的过程，把 Java 源程序预先编译成一种解释器能够识别的格式，再解释执行，这和 Basic 不一样。在这个阶段，我们需要一个 Java 编译器。源文件经过编译后生成以 .class 为后缀的同名文件。编译有两个作用，一是可检查程序的语法错误，二是可在此阶段引入 Java 类库中的类。在 JCreator Pro 中，可以单击【Build】/【Compile file】命令来完成编译工作。

(3) 解释执行程序

在这个阶段，需要一个 Java 解释器，如果编写的是 Java 小程序，它可以是浏览器，如果编写的是应用程序，就需要 java.exe。在 JCreator Pro 中，可以单击【Build】/【Execute file】命令。

2. 建立 Java 运行环境

到此，我们知道了 Java 的运行过程，并看到第(2)、(3)阶段都需要一些工具，JCreator Pro 本身并不提供编译源代码和解释执行程序的工具，而是通过调用 Java 开发工具 JDK 来实现，所以在使用 JCreator Pro 练习编程之前，应在机器上建立起 Java 的运行环境，即在安装 JCreator Pro 之前，应先安装 JDK。

初学者所需的工具都可以从 Sun 公司提供的 JDK 中找到，可到 java.sun.com 网站上去下载最新的 JDK。本书以 JDK 的 j2sdk 1.4.1_02 版为例，将 JDK 安装到合适的目录下，如 C:\j2sdk 1.4.1_02 目录下。

很多情况下我们希望在 Windows 中的任何目录下都能运行 JDK 中的工具，例如把 Java 的源程序放在与 JDK 不同的目录下，如 e:\MyJavaProgram\目录下，并可在该目录下直接运行 JDK 中的工具。要达到这个目的，就要在 Windows 的系统管理工具中设置好环境变量，以 Windows2000 Professional 为例，具体方法如下：

单击【开始】/【设置】/【控制面板】/【系统】/【高级】/【环境变量】，打开【环境变量】对话框，在下面的【系统变量】列表框中设置系统变量，以便以任何用户身份登录都能使用。

如果安装目录是 C:\j2sdk 1.4.1_02，则在【系统变量】列表框选定 Path，单击【编辑】按钮，在 Path 的变量值中加入 C:\j2sdk 1.4.1_02\bin；C:\j2sdk 1.4.1_02\lib 即可，注意和前面的变量值用分号隔开。如图 1-2 所示。

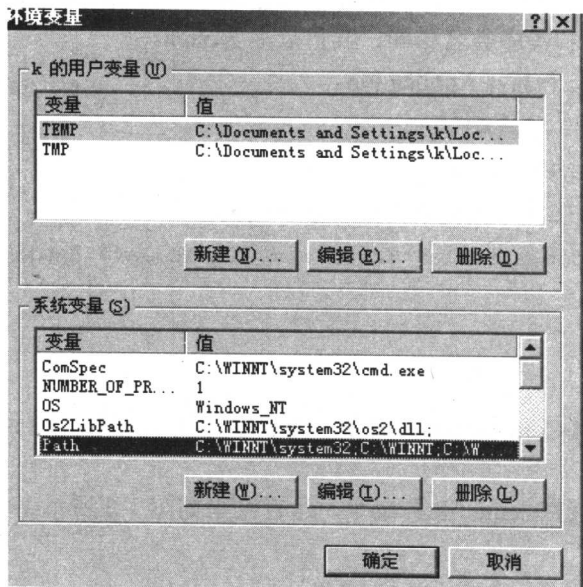


图 1-2 Windows 2000 下的环境变量设置