



华腾教育
HUA TENG EDUCATION

高等院校计算机基础教育规划教材

Java 程序设计

主编 姚南生 副主编 涂兰敬 邢宇飞



WUHAN UNIVERSITY PRESS

武汉大学出版社

内 容 简 介

本书既注重理论的介绍,又强调实际的应用,目的在于培养读者利用面向对象技术和 Java 语言解决实际问题的能力。全书分 10 章阐述,主要内容包括:Java 语言概述,Java 语言编程基础,类和对象,继承、多态和接口,数据类型类和数组对象,Java 类库,Java 数据流,图形用户界面,Java Applet,Java 异常处理等。

本书适合作为高等职业学校、高等专科学校、成人高校、民办高校的教材,也是自学人员的首选用书。

图书在版编目(CIP)数据

Java 程序设计/姚南生主编. —武汉:武汉大学出版社,2006.12

(高等院校计算机基础教育规划教材)

ISBN 7-307-05360-8

I. J… II. 姚… 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(2006)第 145150 号

责任编辑:王金龙

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:wdp4@whu.edu.cn 网址:www.wdp.com.cn)

印刷:北京市昌平百善印刷厂

开本:787×1092 1/16 印张:17.75

字数:428 千字 印数:1~5000 册

版次:2006 年 12 月第 1 版 2006 年 12 月第 1 次印刷

ISBN 7-307-05360-8/TP·227 定价:26.00 元

版权所有,不得翻印;凡购买我社的图书,如有缺页、倒页、脱页等质量问题,请与当地图书销售部门联系调换。

高等院校计算机基础教育规划教材

编委会

主 任：赵云冲

副主任：涂兰敬 王颢媛 杨爽

编 委：（按姓氏笔画排序）

丁青云	王艳梅	王海梅	王 娟
王朝晖	王 辉	方美秀	白海波
邢宇飞	向 蕾	刘少华	刘 辙
孙怀东	孙贤龙	花庆毅	杨希鹏
杨 涛	李运生	李晓洁	李 琴
沈 丹	沈海涛	张国良	张雅辉
陈 曦	金守兵	赵天广	赵红芬
胡 俊	贾云娇	贾 文	钱 勇
徐 楠	殷洪菊	黄 磊	梁 辉
韩丽彦	程灵枝	程灵波	程宗米
廖中华	熊化武		

前言

PREFACE

Java 语言是一种纯面向对象的程序设计高级语言。它具有独立于平台、交互性强、支持线程等强大功能,在编程尤其是网络编程方面得到了越来越广泛的应用和认可,有成为未来主流开发语言的趋势。

本书针对 Java 语言难讲解、难学习的情况,在编写上注重由浅入深、循序渐进的讲解和练习,着重培养读者独立编程的能力和驾驭程序设计语言的能力,以达到培养读者实际运用能力的目的。

本书分 10 章阐述,主要内容包括:Java 语言概述,Java 语言编程基础,类和对象,继承、多态和接口,数据类型类和数组对象,Java 类库,Java 数据流,图形用户界面,Java Applet,Java 异常处理等。

本书适合作为高等职业学校、高等专科学校、成人高校、民办高校的教材,也可作为 Java 初、中级编程人员学习的首选用书。

本书突出计算机科学与技术学科的特点,强调理论与实践结合。用实例讲解的方法避免枯燥、空洞的理论堆砌,引导读者学会实际应用。由于编者水平有限,书中不足和疏漏之处难免,敬请广大读者不吝赐教。

为了配合本书的教学,华腾教育网站 <http://www.huatengedu.com> 提供了免费的电子教案、课后习题参考答案及程序代码,供读者和教学人员下载使用。

编 者

2006 年 11 月

致 读 者

尊敬的老师、同学们：

感谢您使用武汉大学出版社出版的高等院校计算机基础规划教材。为了今后给您提供更加优秀的教材，我们希望您能在百忙之中，抽出宝贵的时间填写下表并寄回我中心，我们将认真研读您的意见并即时提高我们的教材质量。如果您的来信填写认真，并且所提的意见或建议被我们采纳，我们将根据您的需要赠送文化用品一件。

来信请寄：北京 100080—067 信箱 华腾教育教学与研究中心 邮编：100080

或从华腾教育网 <http://www.huatengedu.com> 下载此表格，填写后发送 E-mail 至 ch@dongfangedu.com.cn。

您需要的文化用品：

我们真诚欢迎广大教师来电来信商讨教材的编写和出版事宜。

联系电话：010—62564988 联系人：赵老师

读者反馈表

个人 信息	姓名：	专业：	电话：
	我是： ☐ 老师 ☐ 学生		E-mail：
	校名：		系别：
	职务：		地址：
意 见 与 建 议	书名：		书号：
	我感觉本教材做如下修改将会更好：		
	我觉得本中心的网站如果这样建设会更好：		
	我的其他意见与建议：		



其 他 信 息	我系共有学生_____名。		
	我系领导的姓名、职务及电话： 系主任： 副主任(分管教学)： 副主任(分管后勤)：		
	我的任课老师的姓名、所任课程及电话： 老师： 任教课程： 电话： 老师： 任教课程： 电话： 老师： 任教课程： 电话： 老师： 任教课程： 电话： 老师： 任教课程： 电话： 老师： 任教课程： 电话： 老师： 任教课程： 电话：		
	我系现有专业： ①_____②_____ ③_____④_____		
	我系开设的计算机课程有： ☐ 计算机应用基础 ☐ 计算机文化基础 ☐ 操作系统 ☐ 计算机组成原理 ☐ 计算机组装与维护 ☐ 计算机网络技术基础 ☐ 多媒体制作 ☐ 软件工程 ☐ 数据结构(C语言描述) ☐ 数据结构(C++描述) ☐ 微机原理与接口技术 ☐ 汇编语言 ☐ 单片机原理 ☐ C语言程序设计 ☐ C++程序设计 ☐ 数据库系统原理 ☐ SQL Server 2000/2005 数据库原理与应用 ☐ Visual Foxpro ☐ PowerBuilder ☐ Oracle ☐ Linux ☐ Delphi ☐ Access ☐ Visual C++ ☐ Visual Basic 程序设计 ☐ ASP ☐ Java 程序设计 ☐ Visual Basic. NET ☐ ASP. NET ☐ AutoCAD ☐ 3ds max ☐ Photoshop ☐ 计算机专业英语 ☐ 电子商务 ☐ 电子商务专业英语 ☐ 数字电路 ☐ 模拟电路 ☐ 电路分析基础 ☐ 人工智能 ☐ 微机控制技术 ☐ Internet 技术与应用 ☐ 计算机网络安全与管理 ☐ 局域网组建与组网工程 ☐ 网页设计与制作 ☐ Web 高级程序设计 ☐ 计算机网络原理及维护		
	我系开设的其他课程有： ①_____②_____ ③_____④_____ ⑤_____⑥_____ ⑦_____⑧_____		

如本表格空间不够,请另附信纸!



目 录

CONTENTS

第 1 章 Java 语言概述	1
1.1 Java 的形成与特点	1
1.1.1 Java 的形成	1
1.1.2 Java 的特点	1
1.2 面向对象的基本概念	2
1.3 Java 的开发工具和运行过程	3
1.3.1 Java 的开发工具简介	4
1.3.2 Java 运行环境的安装和配置	6
1.3.3 Java 的编辑、编译和运行	6
本章小结	8
习题 1	8
第 2 章 Java 语言编程基础	9
2.1 数据类型概述	9
2.1.1 基本数据类型	9
2.1.2 常量	14
2.1.3 变量	15
2.1.4 各数据类型之间的转换	18
2.2 运算符与表达式	21
2.2.1 算术运算符及其表达式	21
2.2.2 关系运算符及其表达式	23
2.2.3 逻辑运算符及其表达式	24
2.2.4 位运算符及其表达式	27
2.2.5 其他运算符及其表达式	29
2.2.6 运算符的优先级	31
2.3 流程控制语句	33
2.3.1 if 语句	33
2.3.2 switch 语句	37

2.3.3 for 循环语句	40
2.3.4 while 循环语句	42
2.3.5 跳转语句	44
2.4 方法	50
2.4.1 方法的基本概念	50
2.4.2 参数与返回值	51
2.4.3 递归方法	52
本章小结	54
习题 2	54

第 3 章 类和对象	55
3.1 类的基本概念	55
3.1.1 类和对象	55
3.1.2 定义类	56
3.2 对象的创建与使用	58
3.2.1 创建对象	58
3.2.2 对象成员的使用	59
3.2.3 类成员的访问控制	61
3.3 构造函数	64
3.3.1 构造函数的定义	64
3.3.2 默认构造函数	66
3.3.3 构造函数的使用	67
3.4 参数的传递和 this 的引用	70
3.4.1 对象作为方法的参数	70
3.4.2 关键字 this 的引用	71
3.4.3 类的封装	72
3.5 包	74
3.5.1 包的概念	74
3.5.2 定义包	75
3.5.3 包的引用	76
3.6 类应用实例	79
本章小结	81
习题 3	81

第 4 章 继承、多态和接口	82
4.1 继承和多态的基本概念	82
4.2 类的继承	82
4.2.1 继承的实现步骤	83
4.2.2 属性和方法的继承	83
4.2.3 父类与子类对象的转换	83

4.2.4 构造函数的继承	84
4.3 类成员的覆盖	86
4.3.1 覆盖的使用	86
4.3.2 对被覆盖成员的操作	87
4.4 方法重载	90
4.4.1 方法的重载	91
4.4.2 构造函数的重载	92
4.5 抽象类和最终类	94
4.5.1 抽象类	94
4.5.2 最终类	96
4.6 接口	97
4.6.1 接口的基本概念	97
4.6.2 接口的实现	98
本章小结	100
习题 4	100
 第 5 章 数据类型类和数组对象	 101
5.1 数据类型类	101
5.1.1 数据类型类的属性	101
5.1.2 数据类型类的常用方法	101
5.2 数组	102
5.2.1 数组的定义与创建	102
5.2.2 数组元素的访问	103
5.2.3 二维数组的使用	107
5.3 命令行参数	111
本章小结	112
习题 5	112
 第 6 章 Java 类库	 113
6.1 Java 类库的常用包	113
6.2 java.lang 的常用类介绍	114
6.2.1 String 类和 String Buffer 类	114
6.2.2 System 类	121
6.2.3 Math 类	125
6.3 java.util 的集合类介绍	127
6.3.1 Vector 类	128
6.3.2 Stack 类	130
6.3.3 Hashtable 类	131
6.4 Java 技术文档的使用方法	133
本章小结	134

习题 6	134
第 7 章 Java 数据流	135
7.1 Java 数据流概述	135
7.1.1 流的基本概念	135
7.1.2 java.io 包中的流类	136
7.1.3 InputStream 与 OutputStream 类	137
7.1.4 具体输入/输出流	138
7.2 文件操作类	138
7.2.1 文件操作类的构造函数	138
7.2.2 文件操作类的常用方法	139
7.3 文件输入与输出	142
7.3.1 FileInputStream 类和 FileOutputStream 类的使用	142
7.3.2 对文件中基本数据类型的操作	144
7.3.3 随机文件的读取	147
7.4 标准输入和输出	154
7.4.1 标准输入对象	154
7.4.2 标准输出对象	154
7.4.3 数据类型的转换	156
本章小结	159
习题 7	159
第 8 章 图形用户界面	160
8.1 Java 图形用户界面概述	160
8.1.1 AWT 和 Swing 简介	161
8.1.2 组件和容器	164
8.2 常用组件简介	164
8.2.1 按钮和标签	165
8.2.2 复选框和单选按钮	170
8.2.3 滚动面板	176
8.2.4 文本框	177
8.2.5 列表框	184
8.2.6 对话框	190
8.2.7 菜单	193
8.3 布局管理器的应用	199
8.3.1 布局管理器概述	199
8.3.2 流布局及其使用	199
8.3.3 边界布局及其使用	201
8.3.4 网络布局及其使用	203
8.3.5 卡片布局及其使用	204

8.4 Java 事件处理机制	206
8.4.1 Java 事件处理概述	207
8.4.2 Java 常用事件	208
8.4.3 事件适配器	215
本章小结	217
习题 8	217
 第 9 章 Java Applet	218
9.1 Java Applet 简介	218
9.1.1 Java Applet 示例	218
9.1.2 Java Applet 的运行方式	220
9.2 Java Applet 的基本用法	223
9.2.1 Java Applet 包简介	223
9.2.2 Java Applet 的生命周期	226
9.3 Java Applet 的参数设置	228
9.4 利用 Applet 显示图像和播放声音	230
9.4.1 显示图像	230
9.4.2 播放声音	233
本章小结	236
习题 9	237
 第 10 章 Java 异常处理	238
10.1 异常处理概述	238
10.1.1 异常简介	238
10.1.2 异常处理机制	239
10.1.3 异常的种类	240
10.1.4 错误的种类	244
10.2 异常的处理方法	245
10.2.1 try/catch/finally 的使用	245
10.2.2 异常的声明	254
10.2.3 抛出异常	255
10.2.4 自定义 Java 异常子类	262
本章小结	268
习题 10	268

第 1 章 Java 语言概述

本章要点

- ◆ Java 的形成与特点
- ◆ 面向对象的基本概念
- ◆ Java 的开发工具和运行环境
- ◆ Java 的编辑、编译和运行过程

1.1 Java 的形成与特点

1.1.1 Java 的形成

Java 语言是由美国 Sun 公司创造和发展的一门完全面向对象的程序语言。由于 Java 语言独特的运行机制、对网络编程支持的便利、方便的平台可移植能力及其他优点,使得 Java 语言从诞生到现在短短的十几年时,得到了广大程序员和用户的青睐,在商业应用上取得了很大的成绩。数据显示,目前全球的 Java 开发人员大约为 450 万,而且未来几年这个数字还将飞速增长。

Java 语言的发展历史最早可以追溯到 1991 年,当时 Sun 公司启动了绿色项目(Green Project),着手发展消费电子产品以及其他的软件开发,开发小组主要成员有 James Gosling、Mike Sheridan 和 Patrick Naughton 等,项目开发的时候除了采用 C 和 C++ 语言,小组自己还研发了 Oak 语言(名字源于办公室窗外的一棵橡树),由于商标的冲突,Oak 随后被改称 Java。虽然绿色项目的发展因为种种原因而搁浅,但 Java 却没有随之被埋没。

1994 年,万维网发展势头迅猛,Gosling 认为市场需要一种不依赖实际硬件和软件环境,安全可靠可交互的浏览器,于是使用 Java 语言开发的 Web 浏览器的项目正式启动。1994 年秋天,WebRunner 开发成功;1995 年 5 月 23 日,WebRunner 被正式改名为 HotJava 发表后,业界为之而轰动,Java 语言也随之得以广泛应用,并在编程语言大家族中开始占据一席之地。

1.1.2 Java 的特点

与 C/C++ 等语言相比,Java 给开发人员提供了更简单的语法。例如,完全的面向对象使得开发人员从设计开始就必须采用面向对象的软件设计方法;独特的运行机制使得 Java 语言有着天然的可移植性;取消指针虽然损失了一些编程的灵活性,但带来的却是质量更高的代码。这些优点使得 Java 迅速被开发人员接受,同时也使 Java 自身得到了发展。

独特的设计思想,使得 Java 语言从一开始就拥有大多数其他语言所没有的优点。

• 简单易用

Java 语言的语法相对简单,因此,编写 Java 程序较为容易。

• 面向对象

Java 语言是一门完全面向对象的语言,在 Java 开发过程中面向对象的设计思想贯穿始终。

20 世纪 70 年代初,C 语言应运而生了。

1978 年,由美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的《C 语言程序设计》一书。后来由美国国家标准协会(American National Standards Institute)在此基础上制定了一个 C 语言标准,于 1983 年发表,通常称之为 ANSI C。

目前,从微型计算机到大型计算机都支持 C 编译程序。C 编译程序不仅能在 UNIX 操作系统下运行,而且能在 DOS、Windows 和 Linux 操作系统下运行。由于 C 语言本身具有的优越性,它已经成为在各种计算机上,从系统软件设计到工程应用程序开发都能使用的一种高级程序设计语言。

1.2.2 C 语言的特点

C 语言之所以广泛流行且深受用户青睐,是因为它以其简洁、紧凑、灵活、表达能力强、产生的目标代码质量高和可移植性好等为基本特色,归纳起来有以下主要特点:

1. C 语言具有结构化的控制语句

C 语言提供了结构化程序所必需的基本控制语句,如条件判断语句和循环语句等,实现了对逻辑流的有效控制。C 语言的源程序由函数组成,每个函数各自独立,把函数作为模块化设计的基本单位。C 语言的源文件可以分割成多个源程序,进行单独编译后可连接生成可执行文件,为开发大型软件提供了极大的方便。C 语言提供了多种存储属性,通过对数据的存储域控制提高了程序的可靠性。

2. C 语言具有丰富的数据结构类型

C 语言除提供整型、实型、字符型等基本数据类型外,还提供了用基本数据类型构造出的各种复杂的数据结构,如数组、结构、联合等。C 语言还提供了与地址密切相关的指针类型。此外,用户还可以根据需要进行自定义数据类型。

3. C 语言具有丰富的运算符

C 语言提供了多达 34 种运算符,运算能力十分丰富,它把括号、逗号、赋值等都作为运算符来处理,从而使 C 的运算类型极为丰富,可以实现其他高级语言难以实现的运算。丰富的数据类型与丰富的运算符相结合,使 C 语言的表达力更具灵活性,同时也提高了执行效率。

4. C 语言简洁、紧凑,使用方便、灵活

程序书写自由,共有 9 种控制语句。ANSI C 规定 C 语言共有 32 个关键字。关键字就是已被编程语言本身使用的标识符,不能作变量名、函数名等其他用途。

C 语言的关键字详见附录 I。

注意:在 C 语言中,关键字都是小写的。

不同的实现对 C 语言的关键字有不同的扩充,Turbo C 扩充了 11 个关键字,如表 1-1 所示。

表 1-1 Turbo C 扩充的关键字

asm	_cs	_ds	_es	_ss	cdecl
far	huge	interrupt	near	pascal	

相关操作组织在一起,使其相互关系隐藏在程序体的内部,而对外只表现为与其他封装体之间的接口,这样就可以构造出具有独立意义的软件实体。

在面向对象的程序开发中,每个类中都封装了相关的数据和操作,抽象数据类型就是用“类”这种面向对象工具的可理解和容易操纵的结构来实现的,这样可以降低开发过程的复杂性,提高软件开发的效率和质量。在实际的开发过程中,类可以构建系统内部的模块,由于类的封装性把类内的数据保护得很紧密,模块与模块之间只能通过严格控制的操作接口来相互交流,使它们之间的耦合和交叉大大降低。

封装是一个简单而有效的思想,它的优点包括:

- 信息隐蔽,对外是一个接口单一、功能明确、可在各种适合的环境下都能独立工作的有机单元。

- 模块化,对内则是一个结构完整、可进行高度集中、自我管理、自我平衡的整体。

面向对象的程序设计实现了对象的封装,实现了信息隐藏和模块化,有利于程序的安全性和可移植性,同时也有利于对复杂对象的管理。通过对对象的封装,使得用户不必关心诸如对象的行为是如何实现的这样一些细节问题。

类的封装性使得代码的可重用性大大提高,这样的有机单元特别适合构建大型标准化的软件系统,能够明显提高开发效率。

在面向对象的程序设计中,继承(Inheritance)是十分重要的特征之一。由继承而得到的类为子类(subclass),被继承的类为父类或超类(superclass),父类包括所有直接或间接被继承的类。一个父类能够同时拥有多个子类。但是,因为 Java 不支持多重继承,所以一个类只能有一个直接父类。因此,父类实际上是所有子类的公共域和公共方法的集合,而每一个子类则是父类的特殊化,是对公共域和方法在内涵、功能方面的延伸和扩展。

采用继承的机制来设计、组织系统中的类,可以提高程序的抽象程度,使之更接近于人类的思维方式,同时也可以通过继承较好地实现代码重用,提高软件开发效率,降低系统维护工作量。子类继承父类的状态和行为,同时也可以修改父类的状态或重载父类的行为,并添加新的状态和行为。

继承的优点包括:可以将大问题分解成不同的组件,公共部分可以放入一个父类中,其余可以写在子类中;重用现存的代码,可以节省时间;现存的代码已经测试过并且成功地工作,可以减少错误。

多态(Polymorphism)是指在程序中使用相同的名称来表示不同的含义,它是面向对象程序设计的另外一个重要的特征。例如在同一个类中可有许多同名的方法,但其参数数量和类型不同,完成的操作也不同,这些同名不同参数的方法即可构成多态。

除了使用相同的名称来表示不同的方法以外,多态在面向对象程序设计中还拥有其他含义,如虚方法调用(Virtual method invoking),也称为方法调用绑定(Method callbinding),可以使同样的访问代码适用于所有的子类,这样可以使程序的抽象程度明显提高,也使代码更加简洁规范。

1.3 Java 的开发工具和运行过程

Java 不但为程序员和最终用户提供了一个免费的 Java 软件开发工具,而且还提供了一个特征丰富的语言和运行环境。

为了方便程序员更好地制作并更快运行其程序,Java 提供了扩展类库,这些类库代表了面向对象程序设计的核心部分。Java 类库是一组预先开发的程序码,它可以与单独的应用程序相链接。Java 类库给程序员提供了一个彻底经过测试的健壮性很好的类集合。通过使用这些类库,程序员就不需要重新编写这些代码,大大缩短了开发周期。

1.3.1 Java 的开发工具简介

J2SE SDK(Software Development Kit)通常简称 JDK,是 J2SE 平台下的软件开发包,是 Java 平台中面向一般应用程序开发的平台。

下面是一些基本的 Java 工具,如下所示:

- javac 编译器
- java 解释器
- jdb Java 语言调试器
- javadoc API 文档管理器
- javah 头文件生成器
- appletviewer 小应用程序浏览器
- javap 类文件反汇编器

这些文件包含在/java/bin/目录中,可以在任何目录中运行,前提是在设置运行程序的系统中设置了路径。

1. javac——Java 语言编译器

java 指令用来将 Java 源代码编译成字节码,这是 Java 语言最重要的 JDK 工具之一。

用法:

```
javac [ options ] sourcefiles [ @files ]
```

其中:

options:命令行选项([]表示此项可选)。

sourcefiles:一个或多个(用空格分隔)要编译的源文件(例如 Hello.java)。

@files:一个或多个对源文件进行列表的文件。

参数可按任意次序排列。

2. java——Java 语言解释器

java 指令用来运行 javac 编译的字节码。

用法:

```
java [ options ] class [ arguments ]
```

options:命令行选项。

class:要调用的类名(例如:HelloWorld)。

arguments:传给 main 函数的参数。

3. jdb——Java 语言调试器

用法:

```
jdb [ options ] class [ arguments ]
```

其中:

options:命令行选项。

class:要调试的类名。

arguments:传给 main 函数的参数。

4. javadoc——Java API 浏览器

javadoc 可以产生一组 HTML 格式的相关文件来说明编写的 Java 程序代码。

用法:

```
javadoc [ options ] [ packagenames ] sourcefiles [ @files ]
```

其中:

options: 命令行选项。

packagenames: 一个或多个(用空格分隔)包的名字,不允许使用通配符,如(*)。

sourcefiles: 一个或多个(用空格分隔)要编译的源文件。

@files: 一个或多个对源文件进行列表的文件。

5. javah——头文件生成器

javah 程序可以创建 C 头文件和存根文件,把本地 C 成员函数包入 java 类时,这些文件需要被用到。其中,被创建的头文件给出了有关 java 类的信息,这些信息是 C 成员函数与 java 类交换数据所必需的;存根文件将用来创建将定义 java 对象的结构与 java 对象本身数据相联系的 C 文件。

用法:

```
javah [ options ] classes
```

其中:

options: 命令行选项。

class: 要调用的类名。

6. appletviewer——JavaApplet 浏览器

appletviewer 提供了一个可测试小应用程序 Applet 的 Java 运行环境,appletviewer 读取包含小应用程序的 HTML 文件,并在一个窗口中运行它们。

如果在 javac 编译 Java 程序代码成为字节码后,再建立一个 HTML 文件,使用 appletviewer 执行此 HTML 文件,这种程序称为 JavaApplet。前面讲过,javac 编译 Java 程序代码成为字节码,然后,使用 java 执行字节码类,这种程序称为 Java Application。

7. javap——Java 语言反编译器

javap 命令可以反汇编一个 java 字节代码文件,返回有关可变部分和成员函数的信息。

用法:

```
javap [ options ] classes
```

其中:

options: 命令行选项。

class: 要调用的类名。

在不同操作系统下,安装的 JDK 内容都会略有不同,为了避免混淆,下面统一以 Windows 下的 JDK 为例说明。不同的 JDK 在性能和提供的类库上有些差异,但使用上还是一致的。Sun 制定了 Java 的各个标准规范,也提供了完整的 JDK 下载,其他厂商也根据规范进行了自己的实现,例如 IBMJDK、BlackdownJDK 等。同样,以后的内容是以 Sun 的 JDK 为主。

从推出到现在,SunJDK 经历了几个大的版本更新,从最初的 V1.0 发展到最近的 V1.4.2_02。其中 V1.2 的推出是一个里程碑,并正式使用名称 Java2。自 V1.2 起,Swing 开始作为 JDK 的默认类库出现;V1.3 是对 V1.2 的一个改良;V1.4 除了性能的大幅度提升

外,同时调整了一些标准类库中的 API,例如增加了 NIO 等。对于一般的学习和应用,建议使用 V1.3 以上的 JDK。

1.3.2 Java 运行环境的安装和配置

以 Windows XP 操作系统配置 Java 运行环境为例,首先需要下载 Windows 版的 J2SE SDK,可从 <http://java.sun.com/j2se> 下载所需的版本。下面以 V1.4.0 为例做简单介绍。

下载后需要进行安装,安装过程非常简单,步骤如下:

(1)双击安装文件,在出现的初始界面中单击【Next】按钮,出现许可协议窗口,单击【Yes】按钮接受协议。

(2)选择安装目录,默认的安装目录是 c:\j2sdk\1.4.0,也可以单击【Browse】按钮重新选择安装目录,本书统一采用 c:\java 作为 J2SE 的安装目录。继续单击【Next】按钮。

(3)选择安装组件,如果磁盘空间足够,尽量安装所有组件。其中【Program Files】一项必须选择。

(4)单击【Next】按钮,在出现的窗口中选择使用 Java Plug-in 的浏览器,我们将在浏览器中使用 Java Plug-in 来运行 Java Applet。

(5)单击【Next】按钮,完成安装。安装完毕后,在 c:\java\bin 目录下可以看到 JDK 经常使用的工具。通过设置环境变量,可以方便使用安装目录下 bin 子目录中的各种工具。由于 J2SE 的编译和运行多在 DOS 窗口进行,为了能在任何目录中使用 c:\java\bin 目录下的工具和 Java 类库,可在 Windows 9X 操作系统的系统盘根目录的 autoexe.bat 文件中加入如下语句:

```
set classpath=.;c:\java\lib
path= c:\java\bin
```

设置 JDK 路径后,需要重新启动计算机,以便让 autoexe.bat 文件设置生效。以后,在任何目录下,都可以执行 Java 源代码了。

注意:Windows 2000 以上版本的操作系统,可以打开【控制面板】窗口,双击其中的【系统】图标,在【系统特性】窗口中单击【高级】选项卡,然后单击【环境变量】按钮。在弹出的【环境变量】窗口中,在【系统变量】框中选择【Path】项,单击【编辑】按钮。最后在弹出的【编辑系统变量】窗口的【变量值】框后加上上面两条语句。

1.3.3 Java 的编辑、编译和运行

按照 1.3.2 设置完毕后,就可以在命令行下直接使用 %JAVA_HOME/bin 目录下的 JDK 相关命令。

javac 命令,用来编译 Java 程序,使用格式如下:

```
javac <选项> <类> <参数>
```

编译少量的程序文件时,可以直接在命令行中输入相关命令。如果需要编译的文件很多,可以将待编译的文件名写在一个文件中,假设文件名是“files.lst”,就可以使用命令“javac@files.lst”进行编译。

此外,javac 还可以使用如下参数:

- classpath 为需要编译的程序指定额外的类库
- d 指定编译后的文件输出路径
- g 在目标代码中生成所有的调试信息