

计算机职业教育丛书

Java 程序设计

教程与实训

侯俊杰 编著



科学出版社
www.sciencep.com

计算机职业教育丛书

Java 程序设计教程与实训

侯俊杰 编著

科学出版社

北京

内 容 简 介

Java 语言目前已经成为面向对象开发的主流语言之一, 应用在各个领域。本书是以 Java 2 技术为背景的基础教程, 共分为 14 章, 主要讲述了 Java 2 的基本语法概念和面向对象的编程思想。本书的一大特色就是利用大量的实例向读者演示 Java 的编程方法和编程思想, 语言通俗易懂、流畅生动。本书的开发环境是 Sun 公司提供的 JDK1.4.2, 在 Sun 公司的网站上可以免费下载。

本书可供大专院校学生、各种培训机构的学员、网络技术开发人员, 以及希望学习面向对象技术和 Java 实现技术的人员学习参考。

图书在版编目 (CIP) 数据

Java 程序设计教程与实训/侯俊杰编著. —北京: 科学出版社, 2005
(计算机职业教育丛书)

ISBN 7-03-015470-3

I. J… II. 侯… III. JAVA 语言 - 程序设计 - 职业教育 - 教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 044003 号

责任编辑: 吕建忠 陶宇辰/责任校对: 耿 耘
责任印制: 吕春珉/封面设计: 耕者设计工作室

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>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2005 年 5 月第 一 版 开本: 787 × 1092 1/16

2005 年 5 月第一次印刷 印张: 22 1/2

印数: 1-4 000 字数: 486 000

定价: 29.00 元

(如有印装质量问题, 我社负责调换<环伟>)

销售部电话 010-62136131 编辑部电话 010-62138978-8004 (B101)

前 言

Java 语言自 Sun 公司于 1995 年 5 月正式发布以来,经历了出生、成长、成熟壮大几个阶段,逐渐发展成为 IT 领域里的主流计算模式,从高性能计算到移动计算(主要是手机通信)、智能卡,Java 无处不在。近几年,Sun 公司先后推出的 J2SE、J2EE、J2ME 三大平台开发工具,已经成为软件界开发人员非常欢迎的工具。

目前的统计资料表明,全球至少有 85% 的大型企业正在使用 Java 语言开发自己的应用系统,“Java 进入主流计算模式”已经被业界所公认。

Java 语言的功能强大,是继 C 和 C++ 之后的现代化程序设计语言。Java 类似于 C++,是一种完全面向对象的编程语言。虽然 Java 的很多语义特性与 C 和 C++ 相同,但它也进行了很多改进,其中包括垃圾回收、数组边界检查和强制类型转换等,这为程序的编写提供了很大的灵活性。另外,Java 提供的标准类库为分布式编程、多线程处理和图形用户界面提供了独立于平台的支持。

本书由浅至深,不仅方便初次接触 Java 的初学者学习,还为有一定编程经验的读者提供了相应的指导。本书从面向对象的技术出发,涉及图形用户界面、网络通信、组件、Web 应用等先进应用技术。

本书包含了最新的 Java 编程技术,每章均有很多程序范例,使用者可以边学边练,逐步加深和完善对核心技术的理解。本书所有的程序都在 JDK1.4.2 环境下编译通过并成功运行。

全书共分为 14 章,其中第 1~3 章介绍 Java 语言的基本语法、语义结构;第 4、5 章介绍 Java 语言面向对象的特性;第 6 章介绍数组和字符串;第 7 章介绍异常处理;第 8 章介绍 Java 的输入/输出及文件操作;第 9 章介绍 Java 多线程编程;第 10 章介绍 Java 小程序;第 11、12 章介绍 Java 应用程序的图形化界面设计及组件;第 13 章介绍网络编程;第 14 章介绍 Java 的 Web 应用技术 JSP。

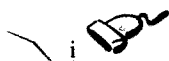
计算机是注重实践的学科,尤其是对编程语言的学习,更应注重实际操作。本书提供了大量的实例,以便读者在学习的过程中巩固、强化所学的知识。在学习的过程中,读者应尽量将所有实例的程序都编译运行一遍,这样可提高编程的实际操作能力。

本书的程序源代码可到 www.tqbooks.net 下载专区下载。

鉴于编者的水平有限,错误之处在所难免,敬请广大读者批评指正。

编者

2005 年 4 月



目 录

第 1 章 Java 语言概述	1
1.1 Java 语言的起源与特点.....	2
1.1.1 Java 语言的起源与发展.....	2
1.1.2 Java 语言的特点.....	3
1.2 Java 开发平台简介.....	4
1.2.1 Java 虚拟机.....	4
1.2.2 JDK 的安装与使用.....	5
1.3 Java 语言程序举例.....	8
1.4 思考与练习.....	12
第 2 章 基本数据类型、运算符与表达式	13
2.1 标识符与关键字.....	14
2.1.1 标识符.....	14
2.1.2 关键字.....	14
2.2 变量与常量.....	15
2.2.1 变量.....	15
2.2.2 常量.....	15
2.3 基本数据类型.....	16
2.3.1 整型.....	16
2.3.2 浮点型.....	17
2.3.3 字符型.....	17
2.3.4 布尔型.....	18
2.3.5 数据类型常量.....	18
2.3.6 类型转换.....	20
2.3.7 各种数据类型的定义与使用举例.....	21
2.4 运算符与表达式.....	22
2.4.1 算术运算符及其表达式.....	22
2.4.2 关系运算符及其表达式.....	23
2.4.3 布尔逻辑运算符及其表达式.....	24
2.4.4 位运算符及其表达式.....	25
2.4.5 赋值运算符及其表达式.....	26
2.4.6 条件运算符及其表达式.....	28
2.4.7 各类运算符的优先级.....	29
2.4.8 综合举例.....	29



2.5	思考与练习.....	31
第 3 章	程序的流程控制.....	32
3.1	分支语句.....	33
3.1.1	if 语句.....	33
3.1.2	if...else 语句.....	34
3.1.3	switch 语句.....	35
3.2	循环语句.....	37
3.2.1	while 语句.....	37
3.2.2	do...while 语句.....	38
3.2.3	for 语句.....	39
3.3	跳转语句.....	41
3.3.1	break 语句.....	41
3.3.2	continue 语句.....	42
3.3.3	return 语句.....	43
3.4	注释语句.....	46
3.5	思考与练习.....	46
第 4 章	类、对象、接口与包.....	48
4.1	面向对象的基本概念与特征.....	49
4.1.1	对象.....	49
4.1.2	类的封装.....	50
4.1.3	类的继承.....	50
4.1.4	类的多态.....	51
4.2	类.....	51
4.2.1	类的声明.....	51
4.2.2	修饰符.....	53
4.2.3	构造函数.....	56
4.2.4	main () 函数.....	56
4.2.5	创建类综合举例.....	57
4.3	对象.....	59
4.3.1	对象的生成.....	60
4.3.2	对象的使用.....	60
4.3.3	对象的清除.....	61
4.3.4	使用对象综合举例.....	61
4.4	接口.....	64
4.4.1	接口的定义.....	64
4.4.2	接口的实现.....	65
4.4.3	使用接口综合举例.....	65
4.5	包.....	71

4.5.1	系统程序包.....	71
4.5.2	包的声明.....	72
4.5.3	导入包.....	72
4.6	思考与练习.....	72
第 5 章	类的继承、多态与内部类	74
5.1	类的继承.....	75
5.1.1	创建子类.....	75
5.1.2	属性与方法的继承.....	76
5.1.3	this 与 super 的使用.....	80
5.2	类的多态.....	81
5.2.1	方法的重载.....	82
5.2.2	方法的覆盖.....	84
5.3	抽象类与方法.....	84
5.4	final 类与 final 成员.....	87
5.5	内部类.....	89
5.5.1	内部类的特性.....	89
5.5.2	静态内部类.....	90
5.5.3	内部类应用举例.....	91
5.6	思考与练习.....	92
第 6 章	数组与字符串	94
6.1	数组.....	95
6.1.1	一维数组.....	95
6.1.2	二维数组.....	98
6.2	动态数组 Vector.....	100
6.2.1	创建 Vector.....	101
6.2.2	访问 Vector 中的对象.....	101
6.2.3	增加或移除 Vector 中的对象.....	102
6.2.4	改变 Vector 的大小.....	102
6.2.5	Vector 应用举例.....	103
6.3	字符串.....	105
6.3.1	String 类.....	105
6.3.2	StringBuffer 类.....	112
6.4	思考与练习.....	115
第 7 章	程序的异常处理	116
7.1	异常处理的基础知识.....	117
7.1.1	异常事件.....	117
7.1.2	异常处理机制.....	118
7.1.3	异常类.....	118

7.2	异常处理方法	120
7.2.1	捕获与处理异常	120
7.2.2	获得异常的有关信息	122
7.2.3	用 throw 抛出异常	124
7.2.4	用 throws 重新抛出异常	125
7.2.5	异常处理综合举例	126
7.3	创建异常类	129
7.4	思考与练习	131
第 8 章	输入/输出流与文件操作	132
8.1	数据流的基本概念	133
8.1.1	理解数据流	133
8.1.2	Java 中的标准数据流	133
8.1.3	Java 包中的数据流	135
8.2	字节流	135
8.2.1	InputStream 与 OutputStream 类	135
8.2.2	文件字节输入流 FileInputStream 类	138
8.2.3	文件字节输出流 FileOutputStream 类	139
8.3	文件处理	140
8.3.1	File 类	140
8.3.2	文件过滤器	145
8.3.3	随机文件操作	147
8.4	字节流的高级应用	151
8.4.1	内存的读/写	151
8.4.2	管道流	153
8.4.3	数据流	153
8.4.4	对象流	156
8.5	字符流	159
8.5.1	Reader 类与 Writer 类	159
8.5.2	字符文件流 FileReader 与 FileWriter 类	160
8.5.3	字符缓冲流 BufferedReader 与 BufferedWriter 类	161
8.6	思考与练习	164
第 9 章	多线程机制	166
9.1	线程与多线程	167
9.1.1	线程与多线程的概念	167
9.1.2	线程的结构	167
9.2	线程的创建	168
9.2.1	创建线程的方法一：继承 Thread 类	168
9.2.2	创建线程的方法二：实现 Runnable 接口	171

9.2.3	线程组.....	173
9.3	线程的控制与调度.....	173
9.3.1	线程的生命周期与状态变迁.....	173
9.3.2	线程的调度机制.....	175
9.3.3	线程的状态转变.....	177
9.4	线程的同步机制.....	179
9.4.1	线程的同步实现.....	179
9.4.2	线程的死锁.....	185
9.5	思考与练习.....	187
第 10 章	编写 Java Applet 程序	188
10.1	Applet 的特点.....	189
10.2	Applet 的生命周期.....	190
10.3	Applet 的图形设计.....	193
10.3.1	Applet 的显示与刷新.....	193
10.3.2	图形坐标系.....	194
10.3.3	显示文字.....	194
10.3.4	控制颜色.....	196
10.3.5	绘制图形.....	199
10.4	Applet 的窗体设计.....	201
10.5	Applet 的事件处理.....	205
10.6	Applet 的多媒体应用.....	206
10.6.1	标识网络上的资源.....	206
10.6.2	显示图像.....	206
10.6.3	播放声音.....	208
10.7	HTML 与 Applet.....	211
10.7.1	Applet 的 HTML 标记属性.....	211
10.7.2	Applet 与浏览器进行通信.....	213
10.7.3	在网页中嵌入 Applet 实例.....	213
10.8	思考与练习.....	216
第 11 章	图形用户界面设计	217
11.1	使用 Swing 创建用户界面.....	218
11.1.1	AWT 与 Swing.....	218
11.1.2	Swing 常用介绍.....	218
11.2	框架与布局管理.....	220
11.2.1	容器与面板.....	220
11.2.2	创建窗口.....	222
11.2.3	创建对话框.....	224
11.2.4	布局管理.....	228



11.3	事件处理	239
11.3.1	Java 中的事件处理机制	240
11.3.2	鼠标事件	241
11.3.3	键盘事件	243
11.3.4	窗口事件	245
11.4	思考与练习	248
第 12 章	Swing 组件库	249
12.1	文本标签	250
12.2	按钮	251
12.3	单选按钮	254
12.4	复选框	256
12.5	列表框	259
12.6	下拉列表	263
12.7	文本框	265
12.8	文本域和滚动条	269
12.9	标尺	271
12.10	菜单	274
12.11	弹出式菜单	281
12.12	工具栏	284
12.13	选项对话框	286
12.14	文件选择对话框	289
12.15	思考与练习	295
第 13 章	Java 网络编程	296
13.1	使用 URL 访问网络资源	297
13.1.1	URL 的概念	297
13.1.2	URL 类	298
13.1.3	URLConnection 类	301
13.2	InetAddress 类	303
13.3	TCP Socket 编程	304
13.3.1	Socket 通信基础	304
13.3.2	TCP Socket 的程序开发	305
13.4	UDP Socket 编程	316
13.4.1	UDP Socket 程序开发	317
13.4.2	多播程序的开发	327
13.5	思考与练习	328
第 14 章	Web 开发技术 JSP	329
14.1	JSP 概述	330
14.1.1	JSP 的优点	330

14.1.2	JSP 容器.....	330
14.1.3	服务器.....	331
14.2	运行 JSP.....	331
14.2.1	JSP 的运行环境 Tomcat.....	331
14.2.2	JSP 的执行过程与生命周期.....	333
14.3	JSP 的基本语法.....	334
14.4	JSP 常用的隐含对象.....	342
14.5	思考与练习.....	348

第 1 章



Java 语言概述



教学目标

了解一门编程语言最好先了解它的背景。本章介绍 Java 的起源及其发展，并概要介绍 Java 语言的特点，本章还会简单介绍 JDK 的安装及使用情况。读者需要掌握如何编译和运行 Java 应用程序和 Java 小程序 (Applet)。



教学提示

目前，Java 已经有多种开发工具，各种工具的功能有所不同，本书中的实例均采用 Sun 公司的 JDK 作为编译、运行工具。读者最好也能安装一个文本编辑工具，例如，功能比较强的 UltraEdit 等。



知识重点

◇ Java 语言的特点

◇ JDK 的安装

◇ 编译运行 Java 程序

◇ 编写自己的小程序



知识难点

◇ Java 语言的特点

◇ 编译运行 Java 应用程序和 Java 小程序

学而不思则罔，思而不学则怠。

——孔子



1.1 Java 语言的起源与特点

1.1.1 Java 语言的起源与发展

【背景知识】

在计算机语言史上,Java 是被公众接受最快、也是普及最快的语言。在它问世后的短短两三年的时间内,世界上已有 150 家大公司向 Java 的创始者 Sun 公司购买了许可证,在不少于 7000 万个平台上使用 Java。

Java 语言是一种新型的面向对象的程序设计语言。1996 年初 Java 语言 1.0 版正式发布后,引起了计算机业界的极大兴趣,在各类杂志、报纸上充满了对 Java 语言的各种评论。

Java 语言诞生于 1991 年,是由 Sun 公司的一个开发小组在开发 Green 项目时完成的。该项目最初是开发一种对于消费类电子产品的与平台无关的软件技术,主要用来解决如电视机、电话、闹钟等家用电器的控制和通信问题的。开始时,该项目组成人员准备采用 C++ 语言,但是他们感觉 C++ 语言太复杂,安全性也差,无法满足项目设计的需要,最后决定基于 C++ 语言开发出一种新的编程语言,开发出来后,最初将其称为 Oak (一种橡树的名字),后来正式改名为 Java 语言。在 1991 年到 1993 年期间,这种语言一直被认为是用来开发消费类电子产品和交互式电视控制器的利器,但是由于市场的原因,Sun 最后放弃了继续开发和研制。但随后 Internet 的异常火爆,使 Sun 看到了 Oak 在计算机网络上的广阔前景,于是通过改造,就产生了 Java。

Java 特别适合于 Internet 应用程序的开发,它的硬件和软件平台无关性直接威胁到了 Windows 和 Intel 的垄断地位,一时间“连 Internet,用 Java 编程”成为编程技术人员的一种时尚。由于 Java 提供了强大的图形、图像、动画、音频、视频、多线程和网络交互的能力,使它在设计交互式、多媒体网页和网络应用方面大显身手,成为当今推广最快的计算机编程语言。1998 年 12 月,Sun 公司发布了 Java 2 平台,JDK 1.2 的发布以及 Java 平台的发布是 Java 发展史上新的里程碑,本书正是基于 Java 2 技术的教程。1999 年 6 月,Sun 公司重新组织 Java 平台的集成方法,并将 Java 企业级应用平台作为发展方向,至今 Java 的大家族中已拥有以下三个主要成员。

(1) J2SE

J2SE (Java 2 platform, standard edition) 包括基本的 Java 2 SDK、工具、运行时 (runtime) 和 API,开发者可以用来编写、部署和运行 Java 应用程序和 Applet 小程序。另外,它还包括早期的 Java 开发工具包,即 Java Development Kit,简称 JDK,例如目前通用的 JDK1.4。

(2) J2EE

J2EE (Java 2 platform, enterprise edition) 建立在 J2SE 的基础上,是 JSP (Java server page)、Servlet、EJB (enterprise JavaBean)、JTS (Java transaction service)、Java Mail 以及 JMS (Java message service) 等多项技术的混合体,并且还包含兼容性测试套件,主要用于开发分布式的、服务器端的多层结构的应用系统,例如电子商务网站。

(3) J2ME

J2ME (Java 2 platform, microedition) 主要用于开发电子产品,例如移动电话、数



字计算顶盒、汽车导航系统等。

本书主要介绍 Java 标准版本 J2SE 的开发知识，若要学习 J2EE 和 J2ME，可查阅相关书籍。

1.1.2 Java 语言的特点

Java 语言是一种简单的、面向对象的、分布式的、解释的、健壮的、安全的、独立于平台的、可移植可扩展的、高性能的、多线程的动态程序设计语言。

1. 简单

Java 的开发者非常谨慎地剔除了许多其他高级语言的不必要特性。例如，Java 不支持有关指针的操作，不支持结构数据类型、操作重载、模板、头文件和多重继承。

2. 面向对象

和 C++ 一样，Java 用类把代码组织成逻辑的模块，运行时程序便从类来创建对象。Java 类能够继承其他的类，但是对多重继承，即一个类从多个类中继承方法或者数据成员，则是不允许的。

3. 分布式

Java 提供了包容广泛的例程库，可处理如 HTTP 或 TCP 这样的 TCP/IP 协议。Java 应用程序可通过一个特定的 URL，打开及访问对象，就像访问本地文件系统那样简单。分布式包括数据分布和操作分布。数据分布是指数据可以分散在网络的不同主机上；操作分布是指把一个计算在不同的主机上进行处理。

4. 解释型

Java 是解释型的语言，它编译后并不生成特定的 CPU 机器代码，而是 Java 字节代码。Java 语言解释器（也称 Java 虚拟机，Java virtual machine, JVM）直接对 Java 字节代码进行解释执行。字节代码本身携带了许多编译信息，使得 Java 语言执行时的连接过程更加简单。通过预先将源代码编译为接近机器指令的字节码，Java 有效地克服了传统解释型语言的性能瓶颈，同时又保持了解释型语言的可移植性。Java 语言解释器可在任何机器上执行 Java 字节码。

5. 健壮性

因为 Java 解释器检查程序中所有的系统存取，所以 Java 程序不会陷于瘫痪。相反地，若发现了严重的错误，Java 程序会创建一个异常，该异常能被捕捉并处理，不会带来任何系统崩溃的危险。

6. 安全性

Java 语言不仅检查所有的内存存取，在网络/分布式环境中还确保在小程序执行时没有病毒的侵入。因为 Java 语言不支持指针，所以程序不能访问未被授权的区域。

7. 平台无关性

Java 依靠 Java 虚拟机实现了平台无关性。Java 应用程序能够在网络的任何地方执行，使其具有了字节代码平台无关性、语言版本完全统一无关性的特征，同时访问底层操作系统功能的扩展类库也不依赖于具体系统。

8. 可移植性

Java 中没有与具体环境有关的概念，其数据类型的大小是固定的，而且 Java 本身环境也具有很强的移植性。

9. 多线程

Java 语言能够包含多线程执行，也就是说，让程序同时执行多个任务。例如，一个多线程程序能够在主线程中接收用户键盘输入的同时，使用另外一个线程在屏幕上显示一幅图像。所有应用程序都必须至少有一个线程，它代表了程序运行的主要流程。

10. 动态特性

Java 语言的设计使它能够适合于一个不断变化的环境。在已有的类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行，并且 Java 语言通过接口来支持多继承，使之比严格的类继承具有更灵活的方式和扩展性，这就是 Java 语言的动态特性，其本身也是面向对象设计方法的扩展。

11. 自动收集无用存储单元

Java 语言有自动垃圾收集功能，也就是说，程序员在程序中不必添加删除占用内存的对象代码，Java 系统可以自动释放已失效的对象所占用的内存。

1.2 Java 开发平台简介

【需求】

Java 中有两类应用程序，一类是自己独立运行入口点的 Java 应用程序（Java application）；另一类是被嵌入在 Web 页面中由 Web 浏览器来控制运行的 Java 小程序（Java Applet）。在执行时，它们都需要 Java 运行系统的支持，对于 Java 应用程序，Java 运行系统一般是指 Java 解释器；而对于 Applet，Java 运行系统一般是指能运行 Applet 的与 Java 兼容的 Web 浏览器，并且其中包含了支持 Applet 运行的环境。

1.2.1 Java 虚拟机

【功能】

Java 程序在编译后生成的不是某种 CPU 的指令码，而是 Java 语言特有的字节代码。在执行时，由于 Java 的目标代码需要 Java 运行系统的支持，虽然 Java 运行系统被建立各种不同的平台上，但为了做到 Java 的可移植性，被建立在不同平台上的 Java 运行系统的功能要求是统一的，为此 Java 引入了 Java 虚拟机，即在机器和编译程序之间加

入了一层抽象的、虚拟的机器。这台虚拟的机器在任何平台上都提供给编译程序一个共同的接口。

Java 字节代码是运行在 Java 虚拟机上的。Java 虚拟机底层的运行系统把字节码转化成实际的硬件调用，这相当于一个 CPU 的功能，因此可以将 Java 虚拟机比做一个小巧玲珑的 CPU。

对于不同的操作系统而言，Java 解释器是不同的，但是 Java 虚拟机的实现都是相同的，这就使得 Java 语言具备了平台无关性。Java 虚拟机由五部分组成：一组指令集、一组寄存器、一个堆栈、一个无用内存单元收集器和一个方法区域，这五个部分是 Java 虚拟机的逻辑抽象成分，不依赖于任何实现技术或组织，但是它们的功能必须是在真实的机器上以某种方式实现。

Java 虚拟机寄存器可保存机器的运行状态，这与微处理器中的寄存器很类似。Java 虚拟机寄存器有四种类型：Java 语言程序计数器、指向操作数栈顶端的指针、指向当前执行方法的执行环境指针、指向当前执行方法局部变量区的第一个变量的指针。Java 虚拟机不定义或使用寄存器来传递或接收参数，而是用堆栈来处理数据传递的，目的是为了**保证指令集的简洁性和实现时的高效性**。

Java 虚拟机的堆栈有三个区域：局部变量区、执行环境区和操作数区。

Java 虚拟机的堆栈是运行时动态分配的对象存储区域。Java 语言可进行自动的无用内存单元的收集，程序无须显示释放占用的内存空间，无用内存单元自动回收的算法依赖于虚拟机最终实现的硬件环境。

Java 虚拟机的方法区是编译后的代码区域，包含方法代码、符号表等。

1.2.2 JDK 的安装与使用

【方法】

在 Sun 公司推出 Java 语言的同时，也推出了 Java 的一系列开发工具，如 JDK (Java developer's kit) 是 Sun 公司推出的 Java 开发工具集。由于 Sun 公司是 Java 语言的创立者和 Java 规范的指定者，所以 JDK 自然是最权威的 Java 开发工具。用户可以从网上免费下载最新的 JDK 版本，其网址是：<http://java.sun.com/downloads.html>。下载完后运行安装文件就可以很容易地完成安装了。安装完成后需要将安装路径 (C:\JDK1.4\bin 或者 D:\JDK1.4\bin，或者其他) 添加到系统根目录下的 autoexec.bat 文件的 path 变量中，其方法有两种 (假设安装路径是 C:\JDK1.4\bin)：

- 通过命令行的方式，即 `set path=%path%; C:\JDK1.4\bin`。
- 在控制面板中设置 path 的值，具体方法是选择“控制面板”|“系统”|“高级”|“环境变量”，在系统变量处找到 path，单击“编辑”，将 C:\JDK1.4\bin 加到后面就可以了。

设置完成后在 DOS 窗口下输入“java”或“javac”回车后，如果出现其方法参数提示信息，则安装正确。如果有问题，还需要重新设置。

提示：通过命令行的方式设置 path 只是一次性的，下次在编译运行 Java 程序时，还需要重新进行设置，而第二种设置方法则是永久性的，建议读者使用第二种方法进行设置。



JDK 提供了丰富的开发工具，其基本工具（basic tools）如表 1.1 所示。

表 1.1 JDK 的基本开发工具

名称	文件名	功能说明
Java 编译器	Javac.exe	将 Java 文件编译成字节码文件
Java 解释器	java.exe	执行编译后生成的字节码文件
Java 文档生成器	javadoc.exe	分析 Java 源文件中的声明和文档注释，并生成关于 Java 源文件的 HTML 页面，在 HTML 页面中描述了公有的和保护的和保护的类、接口、构造方法、方法、域。此外，还会产生一个类的层次结构图
Java 小程序浏览器	appletviewer.exe	用来测试、运行 Java 小程序
Java 压缩工具	jar.exe	Java Archive 文件归档工具，可以把类文件及其他格式的文件，如图形、声音文件等压缩成一个文件，便于程序中引入及网络传输等。Java 中已经提供了一些系统级 jar 文件，如 lib 目录下的 tools.jar、dt.jar 等，用户可以根据需要，将自定义类文件或接口，通过 jar 工具，将编译后的.class 文件打包成 jar 文件，并复制到 JDK 的 lib 目录下以便使用
Java 程序调试器	jdb.exe	调试工具 Debugger 用来帮助用户找到 Java 程序中的错误
C 文件生成器	javah.exe	用于生成 C 语言文件和存根文件
Java 运行时解释器	jre.exe	类似于 Java 解释器，但是并不需要 java.exe 所有与开发相关的选项
类文件分析器	javap.exe	分析编译后的 Java 文件

在这些工具中，最常用的是 javac.exe、java.exe、appletviewer.exe、javadoc.exe 等。下面简单介绍如何使用这些 Java 工具来对 Java 源文件进行编译、运行。

1. 使用文本编辑器输入、编辑 Java 源文件

我们选择 UltraEdit 文本编辑器作为 Java 源文件的编辑工具，UltraEdit 软件的外观如图 1.1 所示。

下面以“Hello World!”程序为例，说明如何编译并运行 Java 应用程序。

【示例 1.1】

```
//HelloWorldApp.java
public class HelloWorldApp{
    public static void main (String []args) {
        System.out.println ("Hello, World!");
    }
}
```

