

面向 21 世纪高等学校计算机类专业规划教材

Java 程序设计教程

曾令明 索 望
甘 刚 宋 波 编著

西安电子科技大学出版社

2007

内 容 简 介

本书从 Java 语言的基本特点入手, 全面介绍了 Java 语言的基本概念和编程方法, 并深入介绍了 Java 的高级特性。全书共 11 章, 涉及 Java 中的基本语法、数据类型、面向对象的三大特征、接口、异常、基础类介绍、输入/输出流、图形界面设计、多线程及网络编程等内容。这些内容基本覆盖了 Java 的实用技术, 是进一步使用 Java 进行技术开发的基础。

本书内容详尽, 实例丰富, 在每章最后均列出了若干习题, 以方便教学。本书适合作为高等学校计算机类专业教材, 也可作为有关专业人员的参考书。

★ 本书配有电子教案, 需要者可与出版社联系, 免费提供。

图书在版编目(CIP)数据

Java 程序设计教程 / 曾令明等编著. —西安: 西安电子科技大学出版社, 2007.10

面向 21 世纪高等学校计算机类专业规划教材

ISBN 978-7-5606-1917-0

I. J… II. 曾… 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(2007)第 139237 号

策 划 李惠萍

责任编辑 殷咸安 李惠萍

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2007 年 10 月第 1 版 2007 年 10 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 17.25

字 数 403 千字

印 数 1~4000 册

定 价 23.00 元

ISBN 978-7-5606-1917-0/TP·0993

XDUP 2209001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜, 谨防盗版。

前 言

Java 语言是美国的 Sun 公司推出的一种令人激动的新型语言，它具有面向对象、平台无关、可移植性强、安全、支持分布式等诸多特征，特别适合开发 Internet 程序，它能使网页中静态的图像和文本元素变得具有动感，栩栩如生。目前，Java 已经成为网络编程语言事实上的标准。

从计算机程序设计语言的发展历史来看，Java 语言是在 C++ 语言之上推出的新一代语言，其语法与 C++ 的语法相似，但剔除了 C++ 中容易出错的成分。Java 增加了诸如内存自动回收之类的新功能，删去了 C++ 中令人费解的、不常用的成分，如运算符重载等。Java 语言自 1995 年问世以来，很快流行于全世界，并获得了极大成功。目前，许多软件项目已选择 Java 语言作为其编程语言，特别是计算机网络方面的软件，如许多大型软件 ERP 就是采用 Java 语言开发的。因此越来越多的人开始学习并使用 Java 语言，全世界已经掀起了一股 Java 热潮。Java 2 的推出使 Java 的用途更加广泛，它适用于各种应用开发，尤其是网络应用和嵌入式系统开发。

本书共分 11 章，分别介绍了 Java 的基本语法、数据类型、语句、类、对象、面向对象的三大特征、接口、异常处理、基础类、输入/输出流、图形界面设计、多线程编程、Java 与网络等内容。

第 1 章介绍 Java 语言的发展、特点、运行机制及运行环境，并以一个例子讲解了 Java 程序的运行步骤。第 2 章和第 3 章介绍 Java 的基础知识，包括数据类型、运算符和流程控制语句。第 4 章和第 5 章涉及类与对象的概念、面向对象的三大特征。第 6 章介绍异常的概念、作用和异常的产生，以及怎样利用 Java 开发环境提供的异常类来处理程序中的异常。第 7 章描述了 Java 基础类库的结构和一些常用类以及如何参考 Java 技术文档。第 8 章通过大量例子演示了 Java 类库中的输入/输出流类的使用。第 9 章详细介绍了 Java 基本的图形编程知识，包括窗口的显示及外观设置、各种组件的使用，还介绍了 Java 的事件处理机制，以及 Applet 的设计。第 10 章讨论了在 Java 应用程序中实现多线程的方法，以及线程间的同步和通信机制。第 11 章描述了 Java 提供的进行网络编程的常用类与接口。

本书在内容安排上根据读者的实际需要，力求浅显易懂，并以丰富的实例来帮助读者理解所学知识。每一个例题都是一个完整的程序，读者可以通过上机验证，加深对知识的理解，同时提高编程水平。每章末均配有精心设计的习题，使读者每学完一章内容后，能够及时检查对该章知识点的掌握程度，并促进读者对知识的灵活应用。

在编写过程中，我们力求写出 Java 的精髓，希望能对读者学习 Java 有所帮助。由于作者知识水平有限，书中不妥之处在所难免，敬请读者批评指正。

编著者
2007 年 8 月

目 录

1 章 概述	1
1.1 程序设计语言	1
1.2 Java 语言简介	2
1.2.1 Java 语言的发展	2
1.2.2 Java 语言的特点	3
1.2.3 Java 运行机制	4
1.3 Java 的运行及运行环境	5
1.3.1 Java 开发工具 J2SDK	5
1.3.2 安装和设置环境变量	5
1.3.3 Java 的编辑、编译和运行	6
1.4 Java 程序分析	8
练习题	9
2 章 语言基础	10
2.1 关键字与标识符	10
2.1.1 关键字	10
2.1.2 标识符	10
2.2 数据类型	11
2.2.1 Java 的数据类型	11
2.2.2 基本数据类型	11
2.3 常量	12
2.3.1 数值常量	12
2.3.2 布尔常量	13
2.3.3 字符常量	13
2.4 变量	13
2.4.1 变量的定义	14
2.4.2 变量的基本操作	15
2.4.3 类型转换	16
2.4.4 数组	18
2.5 运算符与表达式	20
2.5.1 赋值运算符	20
2.5.2 算术运算符与算术表达式	20
2.5.3 关系运算符与关系表达式	23
2.5.4 逻辑运算符与逻辑表达式	23

2.5.5	位运算符.....	25
2.5.6	其他运算符.....	27
2.5.7	运算符的优先级.....	28
	练习题.....	29
3 章	流程控制语句.....	31
3.1	程序的逻辑控制.....	31
3.2	条件选择语句.....	32
3.2.1	if 语句.....	32
3.2.2	switch 语句.....	35
3.3	循环语句.....	38
3.3.1	while 语句.....	38
3.3.2	do-while 语句.....	39
3.3.3	for 语句.....	40
3.3.4	循环语句的嵌套.....	42
3.4	跳转语句.....	42
3.4.1	break 语句.....	42
3.4.2	continue 语句.....	45
3.4.3	return 语句.....	47
	练习题.....	47
4 章	类与对象.....	50
4.1	面向对象技术基础.....	50
4.1.1	面向对象的基本概念.....	50
4.1.2	面向对象的基本特征.....	51
4.2	类的定义.....	51
4.2.1	类头说明.....	52
4.2.2	类体说明.....	52
4.3	对象的创建与使用.....	53
4.3.1	对象的创建.....	53
4.3.2	对象的使用.....	54
4.3.3	对象间的赋值.....	55
4.3.4	对象的清除.....	56
4.4	方法.....	56
4.4.1	方法的定义.....	56
4.4.2	参数传递和返回值.....	60
4.4.3	构造方法.....	66
4.5	修饰符.....	68
4.5.1	访问控制符.....	68

4.5.2	static 修饰符.....	70
4.5.3	其他修饰符.....	74
4.5.4	修饰符的混合使用.....	75
	练习题.....	75
5 章	继承与多态	78
5.1	继承.....	78
5.1.1	继承的概念.....	78
5.1.2	继承的实现.....	80
5.2	多态.....	86
5.2.1	多态的概念.....	86
5.2.2	覆盖.....	86
5.3	重载.....	93
5.3.1	方法的重载.....	93
5.3.2	构造函数的重载.....	94
5.3.3	最终方法.....	96
5.4	抽象方法与抽象类.....	97
5.4.1	抽象方法.....	97
5.4.2	抽象类.....	97
5.5	接口.....	99
5.5.1	接口的定义.....	99
5.5.2	接口的实现.....	101
5.5.3	适配器.....	104
	练习题.....	104
6 章	异常处理	107
5.1	异常的概念.....	107
6.1.1	错误与异常.....	107
6.1.2	异常分类.....	108
6.1.3	错误分类.....	109
5.2	异常的处理.....	110
6.2.1	异常处理机制.....	110
6.2.2	异常处理的方法.....	110
6.2.3	抛出异常(throw).....	118
6.2.4	声明抛出异常(throws).....	119
5.3	自定义异常.....	120
	练习题.....	122

7 章 基础类介绍	125
7.1 包.....	125
7.1.1 包的概念.....	125
7.1.2 构建包.....	125
7.1.3 包的引用.....	126
7.2 Java 类库的结构.....	126
7.3 java.lang 包中的常用类.....	127
7.3.1 Object 类.....	127
7.3.2 Class 类.....	128
7.3.3 System 类.....	130
7.3.4 String 类和 StringBuffer 类.....	133
7.3.5 Math 类.....	142
7.3.6 数据类型类.....	143
7.4 java.util 包中的集合类.....	144
7.4.1 Vector 类.....	144
7.4.2 Stack 类.....	146
7.4.3 Hashtable 类.....	147
7.5 关于 Java 技术文档.....	148
练习题.....	149
8 章 输入与输出流	151
8.1 流的概念.....	151
8.2 字节流类.....	151
8.2.1 字节输入流类.....	151
8.2.2 字节输出流类.....	154
8.3 字符流类.....	158
8.3.1 字符输入流类.....	158
8.3.2 字符输出流类.....	161
练习题.....	163
9 章 图形用户界面设计	165
9.1 概述.....	165
9.1.1 AWT 简介.....	165
9.1.2 Swing 简介.....	166
9.1.3 Swing 和 AWT 的不同之处.....	166
9.2 AWT 图形用户界面设计.....	167
9.2.1 java.awt 包.....	167
9.2.2 组件、容器、布局管理器概述.....	168
9.2.3 常用容器.....	173

9.2.4	布局管理器.....	175
9.2.5	AWT 组件库.....	182
9.2.6	事件处理.....	195
9.3	Swing 图形用户界面设计.....	203
9.3.1	Swing 组件的层次.....	204
9.3.2	Swing 组件的使用.....	204
9.4	Applet 与 HTML.....	209
9.4.1	Applet 简介.....	209
9.4.2	Applet 程序中使用的几个基本方法.....	209
9.4.3	在 HTML 中嵌入 Applet 程序.....	211
9.4.4	Font 类和 Color 类.....	215
	练习题.....	221
10 章	多线程编程.....	223
10.1	线程的概念.....	223
10.1.1	程序与进程.....	223
10.1.2	进程与线程.....	223
10.1.3	线程的结构.....	224
10.2	Thread 类与 Runnable 接口.....	224
10.2.1	Thread 类.....	224
10.2.2	Runnable 接口.....	226
10.3	在程序中实现多线程.....	226
10.3.1	使用 Thread 类创建线程.....	226
10.3.2	实现 Runnable 接口.....	227
10.3.3	关于两种创建线程方法的讨论.....	232
10.4	线程的控制与调度.....	232
10.4.1	线程的生命周期.....	232
10.4.2	线程调度与优先级.....	233
10.5	线程间的同步.....	236
10.5.1	线程同步介绍.....	236
10.5.2	同步代码段.....	238
10.5.3	同步方法.....	239
10.6	线程间的通信.....	240
	练习题.....	242
11 章	网络编程技术.....	243
11.1	网络编程基本概念.....	243
11.1.1	网络基础知识.....	243
11.1.2	TCP/IP 协议.....	244

11.1.3	Java 中的网络支持.....	245
11.2	基于 URL 的网络编程.....	245
11.2.1	URL 介绍.....	245
11.2.2	URL 类.....	246
11.2.3	URLConnection 类.....	249
11.3	基于 InetAddress 的网络编程.....	253
11.3.1	InetAddress 类介绍.....	253
11.3.2	使用 InetAddress.....	253
11.3.3	URL 和 InetAddress 的联合使用.....	254
11.4	基于 Socket 的网络编程.....	255
11.4.1	Socket 介绍.....	255
11.4.2	TCP Socket 技术.....	255
11.4.3	创建 TCP Socket.....	256
11.4.4	简单的 Client/Server 程序设计.....	257
11.4.5	支持多客户的 Client/Server 程序设计.....	260
11.4.6	UDP Socket 技术.....	261
11.4.7	创建 UDP Socket.....	261
11.4.8	用数据报进行广播通信.....	262
	练习题.....	264

第 1 章 概 述

本章首先对程序设计语言作了综述；然后介绍了 Java 语言的特点及其运行机制；最后，介绍了 Java 程序的开发环境和步骤。

学习重点：

- Java 语言的特点
- Java 运行机制
- 建立 Java 的运行环境
- Java 应用程序基本框架

1.1 程序设计语言

语言是用于传达信息的表示方法、约定和规则的集合，是人们交流信息的工具和媒介。而程序设计语言是实现程序设计，以便于人与计算机进行信息交流的工具和媒介。程序设计语言一般分为机器语言、汇编语言和高级语言三种。

1. 机器语言

机器语言是以二进制形式表示的机器指令的集合，是计算机系统不需要翻译就可以直接识别和执行的程序设计语言。用机器语言编写程序，程序人员必须熟悉机器指令的二进制代码，记忆指令代码能完成的操作，还应指出这一操作对象的位置，即记忆指令的操作码和地址码。由于机器语言程序是直接在计算机上执行的，因此效率比较高，能充分发挥计算机的高速计算能力。在计算机发展的初期，人们都是使用机器语言直接编制程序。但机器语言不易记忆和理解且缺乏直观性，所以用机器语言编制程序的难度很大。

2. 汇编语言

汇编语言用助记符来表示指令的操作码和操作对象，用标号和符号来代替地址、常量和变量。例如，“ADD AX,BX;”代表两个寄存器数相加的功能。用汇编语言编写的程序，较直观、易理解。但计算机却不能识别和直接运行汇编语言程序，必须由汇编程序将汇编语言程序“翻译”成机器语言程序后才能识别并运行。此外，不同 CPU 的指令系统对应的汇编语言不同。如：单板机、单片机、微处理器等，随机器型号、类型的不同，各自的汇编语言也不同。这就是说，用汇编语言编写的程序缺乏通用性。

3. 高级语言

高级语言是与机器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读性较好，便于维护。同时，由于高级语言并不直接和硬件相关，其编制出来的程序的移植性和重用性也要好得多。常见的高级语言有 Basic、Pascal、C、C++和 Java 等，现代应用程序设计多数都是使用高级语言。

按照思维方式的不同，高级语言可以分为面向过程的语言和面向对象的语言。

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方法。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。

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。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信和交互，以此构成世界的运转。用计算机专业的术语来说，对象的特点就是它们的属性，而能做的事就是它们的方法。常见的面向对象的程序设计语言包括 C++和 Java 等。

用高级语言编写的程序称为源程序。高级语言源程序的执行方式有两种：一种是“编译执行”，一种是“解释执行”。

编译执行是指源程序需先通过特定的工具软件(编译程序)整体编译成机器语言，然后交给操作系统执行，执行时不需要其他应用程序的支持。常见的编译执行的程序语言有 Pascal、C 和 C++。

解释执行是指源程序不需要整体编译，而是读入一句，解释执行一句，整个过程需要一定的软件环境来做源代码的解释器。常见的解释执行的程序语言有 Basic 和 Java。

1.2 Java 语言简介

1.2.1 Java 语言的发展

Java 语言的前身被称为 Oak 语言，它是由美国 Sun 公司于 1991 年开发的一个称为“Green”的软件项目，该项目的本意是开发一个用于消费类电子产品的与平台无关的软件技术。从那以后，Oak 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工具。1994 年，Sun 公司的两个开发人员在 Oak 的基础上创建了 HotJava 的第一个版本，当时称为 Webrunner，即在 Web 上使用的图形浏览器，经过一段时间后才称为 Java。1995 年 5 月，Sun 公司对外正式发布了 Java 语言，随即得到了各 WWW 厂商的大力支持，纷纷在浏览器上加入 Applet 小程序(用 Java 语言编写的小应用程序)，并通过 Internet 在世界各地进行传播。

Java 语言发展到目前为止，有 4 个主要的版本：Java 1.0(1995 年)，Java 1.1(1997 年)，

带有 SDK1.2 的 Java 2(1998 年)和带有 SDK1.3 的 Java 2(2000 年)。

实际上, Java 语言是和 Internet 同步发展起来的一种新型网络语言, 是近年来计算机软件环境中最有意义的进步之一。Java 语言在网络中的地位同超文本链接标注语言 HTML(HyperText Markup Language)一样重要。Java 语言是一种强有力的网络编程语言, 它最大限度地利用了网络资源, Applet 小程序可以跨平台、跨操作系统、跨网络运行, 由于 Applet 代码小, 因此易于在网络上快速地下载和发送, 且不需要修改应用程序就可增加新的 Web 页。

Java 还为编程技术人员提供了许多公用的系统接口。随着 Internet 在全世界范围内的广泛流行, 以及在各个领域的渗透, Java 语言已被各行各业的人士所接受。

1.2.2 Java 语言的特点

Java 语言之所以被人们广泛认可, 是因为它具有许多先进的技术特点, 主要如下。

1. 移植性好

我们知道, Internet 网上运行着数以万计的各种各样的计算机系统, 这些系统上的计算机软、硬件千差万别。要让应用软件在网络上任何一种计算机系统中都能正常运行, 就必须保证使应用软件的移植性好。也就是说, 应用软件本身不受计算机硬件和操作系统的限制(即与平台无关), 软件代码可以在不同的计算机环境中正常运行。Java 语言就是一种与平台无关、移植性好的编程语言, 在源程序级就保证了其基本数据类型与平台无关。Java 源程序经编译后产生的二进制代码是一种与具体指令无关的指令集合, 通过 Java 虚拟机(Java Virtual Machine, JVM), 可以在不同的平台上运行。

2. 纯面向对象技术

Java 是一种完全面向对象的程序设计语言(Oriented Object Programming, OOP)。Java 语言代码以类的形式组成。Java 删除了 C++ 中非面向对象的特性, 使得应用程序的开发变得十分容易。Java 面向对象技术的特征主要有封装性、继承性和多态性, 本书后续内容将详细介绍。

3. 分布式

Java 语言中提供了一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的类库, 通过这些类库, Java 应用程序可以通过 URL 打开并访问网络上的对象。因此, 用 Java 语言可以很方便地编写适合 Internet 和分布式环境下的应用程序。

4. 安全性

分布式计算环境要求软件具有高度的稳定性和安全性。Java 语言具有强大的安全结构和安全策略。Java 之所以具有高质量的安全性, 是因为采取了以下一些措施:

- (1) 取消了指针操作, 不允许直接对内存进行操作。
- (2) 实现内存管理自动化, 内存布局由 JVM 决定。
- (3) 提供了字节码检验器, 以保证程序代码在编译和运行过程中接受一层层的安全检查, 这样可以防止非法程序或病毒的入侵。
- (4) 提供了文件访问控制机制, 严格控制程序代码的访问权限。

(5) 提供了多种网络软件协议的用户接口,用户可以在网络传输中使用多种加密技术来保证网络传输的安全性和完整性。

5. 编译和解释的结合性

Java 是一种编译和解释相结合的语言。一个 Java 语言源程序要运行,必须先由 Java 编译器编译成字节码(Byte Code),即文件扩展名为 .class。这个字节码文件不是最终的执行程序,不能在具体平台上运行,而必须再由运行系统上的字节解释器将其翻译成机器语言,达到边翻译边执行程序的目的。由于这个解释过程是在编译后的字节码上进行的,因此运行速度仍然相当快。

1.2.3 Java 运行机制

Java 程序的运行必须经过编写、编译、运行三个步骤。

(1) 编写是指在 Java 开发环境中进行程序代码的输入,最终形成后缀名为 .java 的 Java 源文件。

(2) 编译是指使用 Java 编译器对源文件进行错误排查的过程,编译后将生成后缀名为 .class 的字节码文件,这不像 C 语言那样最终生成可执行文件。

(3) 运行是指使用 Java 解释器将字节码文件翻译成机器代码,执行并显示结果。这一过程如图 1-1 所示。

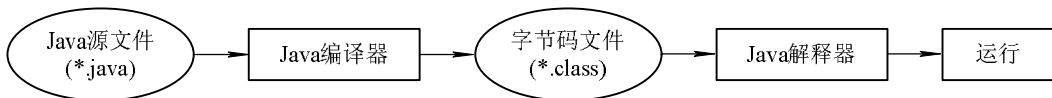


图 1-1 Java 程序运行流程

字节码文件是一种与任何具体机器环境及操作系统环境无关的中间代码,它是一种二进制文件,是 Java 源文件由 Java 编译器编译后生成的目标代码文件。编程人员和计算机都无法直接读懂字节码文件,它必须由专用的 Java 解释器来解释执行,因此 Java 是一种在编译基础上进行解释运行的语言。

Java 解释器负责将字节码文件翻译成具体硬件环境和操作系统平台下的机器代码,以便执行。因此 Java 程序不能直接运行在现有的操作系统平台上,它必须运行在被称为 Java 虚拟机的软件平台之上。

Java 虚拟机(JVM)是运行 Java 程序的软件环境,Java 解释器是 Java 虚拟机的一部分。在运行 Java 程序时,首先会启动 JVM,然后由它来负责解释执行 Java 的字节码,并且 Java 字节码只能运行于 JVM 之上。这样利用 JVM 就可以把 Java 字节码程序和具体的硬件平台以及操作系统环境分隔开来,只要在不同的计算机上安装了针对于特定具体平台的 JVM,Java 程序就可以运行,而不用考虑当前具体的硬件平台及操作系统环境,也不用考虑字节码文件是在何种平台上生成的。JVM 把这种不同软硬件平台的具体差别隐藏起来,从而实现了真正的二进制代码级的跨平台移植。JVM 是 Java 平台无关的基础,Java 的跨平台特性正是通过在 JVM 中运行 Java 程序实现的。

Java 语言这种“一次编写，到处运行(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系统来编译产生不同机器代码的问题(即硬件环境和操作平台的异构问题)，大大降低了程序开发、维护和管理开销。

需要注意的是，Java 程序通过 JVM 可以达到跨平台特性，但 JVM 是不跨平台的。也就是说，不同操作系统之上的 JVM 是不同的，Windows 平台之上的 JVM 不能用在 Linux 上面，反之亦然。

1.3 Java 的运行及运行环境

1.3.1 Java 开发工具 J2SDK

Java 不仅提供了一个丰富的语言和运行环境，而且还提供了一个免费的 Java 软件开发工具集(Java Development Kits, JDK)。到目前为止，Sun 公司先后发布了多个主要的 JDK 的版本。其主要版本的发布时间和特点如下：

(1) JDK 1.0: 1995 年推出，是 Java 最早的编译环境。

(2) JDK 1.2: 1998 年发布，在图形化用户界面、数据库互连以及其他许多方面作了重大改进。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，因此从 Java 1.2 版本后的 Java 语言也称作 Java 2 语言。

(3) JDK 1.3: 2000 年 5 月发布，在多媒体应用、编译速度等方面做了改进。

(4) JDK 1.4: 2002 年 2 月发布，增加了部分功能。

(5) JDK 1.5: 2004 年 10 月发布，改名为 J2SE5.0，是目前最新版本。

通常所说的 J2SDK，是指与 Java 2 语言相对应的 Java SDK(Java Software Development Kits)，它是对较新版本中 JDK 的一个特定称呼。J2SDK(JDK)包括 Java 的编译器、解释器、调试器等开发工具以及 Java API 类库。编程人员和最终用户可以利用这些工具来开发 Java 程序。其调试工具主要包括 Java 语言编译器 javac，用于编译 Java 源程序；Java 字节码解释器 java，用于解释运行 Java 程序，显示程序运行结果；小应用程序浏览工具 appletviewer，用于测试并运行 Java 小程序。

1.3.2 安装和设置环境变量

为了建立起 Java 的运行环境，可以到 Sun 公司的网站(<http://www.sun.com>)上下载最新的 JDK。建议同时下载其 Java Documentation，这是 Java 帮助文档。

以 JDK 1.5 版的 J2SDK 为例(即 j2sdk 1.5.0)，运行下载的 jdk-1_5_0_11-windows-i586-p.exe 文件，开始进行 Java 环境的正式安装。默认安装于 c:\Program Files\Java\jdk1.5.0_11 目录下，用户可以更改这个默认安装目录。

接下来需要设置运行环境，以便能够在 Windows 的任何目录下面都能编译和运行 Java 程序。对于 Windows 2000 或 XP 的用户，使用鼠标右击【我的电脑】，依次选择【属性】 | 【高级】 | 【环境变量】，打开【环境变量】对话框，在【系统变量】列表框中选中 Path 行，如图 1-2 所示。

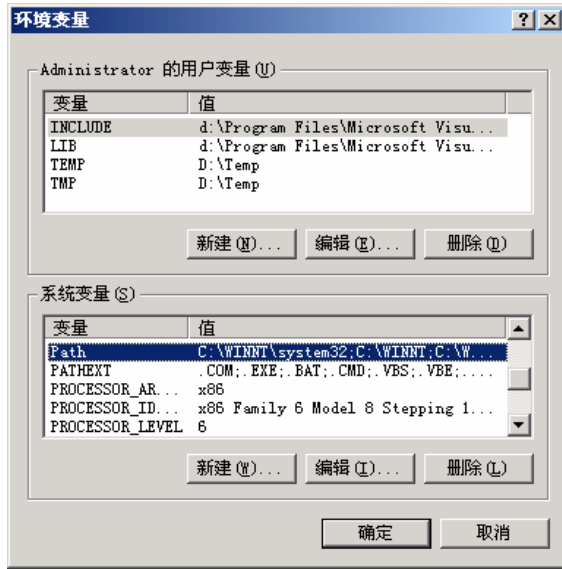


图 1-2 环境变量设置

单击【编辑】按钮，在已有内容的最后增加 JDK 下 bin 和 lib 这两个目录安装的位置，每一项用分号隔开，图 1-3 中最后两项是添加的。这样做的目的是让 Windows 在任何目录下都能运行 JDK 中的工具。读者可以把 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。



图 1-3 Path 变量的设置

1.3.3 Java 的编辑、编译和运行

Java 源程序是一种文本文件，可以使用任何的文本编辑器编写，只是要注意存储时的文件名后缀名必须是.java。建议读者把所有源程序文件都保存在一个指定的目录下，便于调试和运行。

这里向大家推荐两种编辑器：一是 Windows 的记事本，一是文本编辑工具 UltraEdit 和 EditPlus。使用 Windows 记事本编辑 Java 源程序文件，存储时先选择 *.*(所有文件)的文件类型，然后，输入带有 .java 后缀的文件名；或者直接以带英文双引号 "XXXX.java" 的形式输入文件名。UltraEdit 和 EditPlus 是两个非常易用且功能强大的文本编辑工具。编辑时，它们自动地把关键字、常量、变量等不同元素用不同的颜色区分开来，从而有助于减少语法错误。

应当指出,尚其他的 Java 开发工具可供选择,如 JCreator、JBuilder、Eclipse、NetBean、Visual J++ 等。这些开发工具功能强大,但操作复杂,不适合初学者。

Java 是解释型语言。Java 源程序必须先由 Java 编译器进行编译,生成字节码文件(也称类文件),然后在 Java 解释器的支持下解释运行。

Java 编译器是程序 javac.exe,其用法如下:

```
javac filename.java
```

其中 filename.java 是 java 源程序文件的文件名。如果编译器没有返回任何错误信息,则表示编译成功,并在同一目录下生成与类名相同的字节码文件 filename.class。如果编译出错,则需查找错误原因,进一步修改源程序,并重新编译。

Java 解释器是程序 java.exe,其用法如下:

```
java filename
```

其中 filename 是编译生成的 java 字节码文件的文件名,注意不要带后缀名 .class。

下面是一个以 Windows 下的记事本编辑的 Java 应用程序,并在 JDK 下运行的过程。程序内容将在以后章节中予以解释,读者只需熟悉编译及运行过程。

【例 1-1】Java 应用程序示例。

(1) 编写程序:打开记事本,编辑下面的程序,保存到 E:\MyJavaProgram 目录下(先在 E 盘创建好此目录),文件名为“JavaProgram1.java”,注意大小写。

```
public class JavaProgram1 {  
    public static void main (String args [ ]) {  
        System.out.println("I have been a programmer! ");  
    }  
}
```

(2) 编译程序:单击【开始】|【运行】命令,在命令行上输入“cmd”,按回车键,即可打开一个命令窗口,将目录转换到编写的 Java 源程序所在的目录,即 E:\MyJavaProgram,输入“javac filename.java”,本例应输入“javac JavaProgram1.java”,如果编译正确的话,显示如图 1-4 的结果。

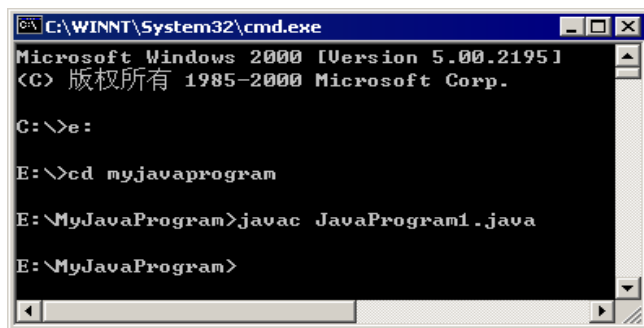


图 1-4 编译 Java 程序

(3) 执行程序:同样在命令窗口中输入“java filename”,本例应输入“java JavaProgram1”,运行结果如图 1-5 所示。

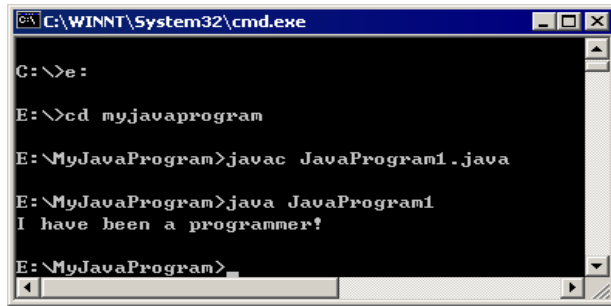


图 1-5 运行 Java 程序

1.4 Java 程序分析

通过前面例 1-1 的 JavaProgram1 程序，可以发现 Java 应用程序基本框架如下：

```
public class 类名 {  
    public static void main ( Stringargs [ ] ) {  
        ...//程序代码  
    }  
    //其他属性和方法  
}
```

在下面的分析中，为了叙述方便，我们特地为 Hello World 程序的源代码行增加了编号。

```
1 /* HelloWorld.java */  
2 public class JavaProgram1 {  
3     public static void main(String args[]) {  
4         System.out.println("I have been a programmer!");  
5     }  
6 }
```

第 1 行是注释行。

Java 语言主要有三种注释：形如“/*注释内容*/”的格式，可以注释一行或多行文本；形如“//注释内容”的格式，可以注释一行文本；形如“/**注释内容*/”的格式，可以注释一行或多行文本，并可用于生成专门的 Javadoc。注释可以放在一行的开头或某个语句之后，为程序增加必要的解释，提高程序的可读性。

第 2 行是类的定义。

使用关键字 class 定义了一个 HelloWorld 类，class 前面的 public 关键字表示这个类的访问特性是公共的。

Java 语言中的基本程序单位是类，一个程序文件可以定义多个类，但仅允许有一个公共的类。源程序的文件名要与公共类的名称相同(包括大小写)，其扩展名为 .java。因此，JavaProgram1 程序的源程序文件名必须是 JavaProgram1.java。

第 2 行最后到第 6 行的一对花括号，其内部是类体。类体中可以声明类的变量(属性)和类的方法(函数)，它们是类的成员。本例中没有声明类的成员变量。