

21 世纪高校规划教材

# JAVA 语言 程序设计教程

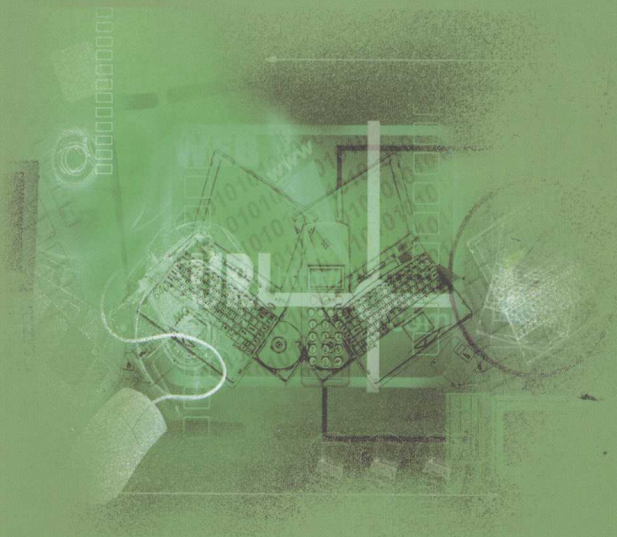
主编 / 潘 杰



江西高校出版社

责任编辑 魏文清

封面设计 王瑞英



## 21 世纪高校规划教材（计算机）书目

《计算机应用基础》（第三版）（非计算机专业公共课教材）（省级优秀教材）

《计算机一级考试强化训练》（非计算机专业公共课实训教材）

《计算机应用基础教程》（非计算机专业公共课教材）

《办公数据处理》（非计算机专业公共课教材）

《网页设计与制作》（省级优秀教材）

《数据库系统原理及应用》（第二版）

《Visual Basic 程序设计教程》（第二版）

《VB 程序设计教程习题解答与上机指导》

《微机组装与维护实用教程》（第二版）

《C 语言程序设计》（省级优秀教材）

《C 语言程序设计习题及实训指导》

《计算机网络技术基础》（第二版）

《计算机专业英语》

《多媒体技术与应用》

《Flash8.0 动画实用教程》

《计算机网络技术实用教程》

《计算机图像处理——PHOTOSHOP CS2 教程》

《计算机图像处理——PHOTOSHOP CS2 实战解析》

《Delphi & SQL Server 管理系统程序设计》

《局域网组建与配置》

《3Ds MAX 8 效果图实用教程》

《JAVA 语言程序设计教程》

ISBN 978-7-81132-288-0



9 787811 322880 >

定价:20.60 元



21 世纪高校规划教材

# Java 语言程序设计教程

主 编:潘 杰

副主编:徐术力 彭 健 刘楚雄

江西高校出版社



## 图书在版编目(CIP)数据

Java 语言程序设计教程/潘杰主编. —南昌:江西高校出版社, 2008.8

ISBN 978 - 7 - 81132 - 288 - 0

I .J... II .潘... 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(2008) 第 119706 号

|         |                                |
|---------|--------------------------------|
| 出版发行    | 江西高校出版社                        |
| 社 址     | 江西省南昌市洪都北大道 96 号               |
| 邮 政 编 码 | 330046                         |
| 总编室电话   | (0791)8504319                  |
| 销 售 电 话 | (0791)8508191                  |
| 网 址     | www.juacp.com                  |
| 印 刷     | 江西龙莹印务有限公司                     |
| 照 排     | 江西太元科技有限公司照排部                  |
| 经 销     | 各地新华书店                         |
| 开 本     | 787mm × 1092mm 1/16            |
| 印 张     | 12                             |
| 字 数     | 292 千字                         |
| 版 次     | 2008 年 8 月第 1 版第 1 次印刷         |
| 印 数     | 1 ~ 2500 册                     |
| 书 号     | ISBN 978 - 7 - 81132 - 288 - 0 |
| 定 价     | 20.60 元                        |

版权所有 侵权必究

# 前 言

Java 语言是由 Sun 公司开发的一个面向对象语言。因为它的简单易用、平台无关性、安全机制、高可靠性,在短短十来年时间里,很快就发展成为当前最流行的编程语言。同时,为了适应企业应用和嵌入式系统开发,Sun 公司推出了相应的 J2EE 和 J2ME 开发工具包与相应规范。目前,它们已经成为各自相关领域内的主流产品。在当今的网络时代,学习 Java,掌握 Java 编程,对程序员来说不再是可有可无,Java 已经成为开发人员必备的工具了。本书讲解 Java 语言的基本知识和应用,这同时也是 J2EE 和 J2ME 应用开发的基础。

本书编者多年从事 Java 语言的教学工作,熟悉学习对象的情况,力求做到突出重点、详析难点、解答疑点,使读者学习起来容易理解和掌握;本书语言通俗,由浅入深,简明实用,适于自学;书中通过大量的例题来解释相关的概念和方法,将明确概念和着重实用相结合,有助于读者对基本概念和方法的学习;本书每章配有一定量的练习题,可引导读者掌握本章内容并检查学习情况,有助于读者较快地掌握所学的知识。

由于 Java 发展很快,我们在开发环境上选择在 Windows XP 操作系统上采用目前最新的版本 jdk1.6.0\_06,来编写调试所有的程序。全书内容共分为九章。第一章主要介绍 Java 的特点和运行环境。第二和第三章主要讲述程序设计的基本语法和程序流程控制。第四章主要讲述面向对象程序设计思想和相关程序设计技术。第五章主要介绍系统常用类和数组这一基本数据结构的使用,以及集合类的应用。第六章主要介绍异常处理的概念和使用方法。第七章主要介绍输入输出流以及网络流的概念和应用。第八章主要介绍多线程技术。第九章主要介绍 Applet 与图形界面编程。

学习任何一种计算机语言都离不开上机实践。通过上机调试程序,可以发现概念上的模糊和方法的错误,搞清概念和纠正错误对于深刻理解和熟练掌握 Java 语言十分重要。敬劝读者学习 Java 语言时,一定要做到边看书边上机,只有多做练习、多上机调试程序,才会较快地掌握这种语言。

本书由潘杰担任主编,徐术力、彭健、刘楚雄担任副主编,参加编写的有钟平、卢镭、薛琛、罗海勇等。在编写本书的过程中得到了许多同行、专家及领导的关心和支持,在此表示衷心的感谢。

由于作者水平有限,编写时间仓促,在该书中可能会出现一些错误,恳请读者提出指正和宝贵意见。

编者

2008.8



# 目 录

|                               |    |
|-------------------------------|----|
| 第一章 Java 初识 .....             | 1  |
| 1.1 Java 的起源与发展 .....         | 1  |
| 1.1.1 Java 的起源 .....          | 1  |
| 1.1.2 Java 语言的主要特点 .....      | 2  |
| 1.1.3 Java 语言的应用发展 .....      | 2  |
| 1.2 Java 程序设计示例 .....         | 3  |
| 1.2.1 Java 程序开发的环境配置 .....    | 3  |
| 1.2.2 编写第一个 Java 应用程序 .....   | 7  |
| 1.2.3 Java 注释的用法 .....        | 9  |
| 1.3 Java 虚拟机 .....            | 10 |
| 1.3.1 传统程序(C 程序)运行的一般过程 ..... | 10 |
| 1.3.2 Java 程序运行的一般过程 .....    | 10 |
| 1.3.3 Java 虚拟机的组成 .....       | 10 |
| 第二章 Java 语法基础 .....           | 12 |
| 2.1 Java 符号集 .....            | 12 |
| 2.1.1 标识符 .....               | 12 |
| 2.1.2 关键字 .....               | 13 |
| 2.1.3 运算符 .....               | 13 |
| 2.1.4 分隔符 .....               | 13 |
| 2.2 数据类型常量与变量 .....           | 14 |
| 2.2.1 数据类型的概念 .....           | 14 |
| 2.2.2 常量 .....                | 14 |
| 2.2.3 变量 .....                | 16 |
| 2.3 表达式和语句 .....              | 19 |
| 2.3.1 算术表达式 .....             | 19 |
| 2.3.2 赋值表达式 .....             | 22 |
| 2.3.3 表达式语句与语句块 .....         | 23 |
| 2.3.4 关系表达式 .....             | 24 |
| 2.3.5 逻辑表达式 .....             | 24 |
| 2.3.6 位运算表达式 .....            | 25 |
| 2.3.7 条件运算符和表达式 .....         | 25 |
| 2.3.8 运算符的优先级 .....           | 25 |
| 第三章 程序流程控制 .....              | 28 |
| 3.1 选择结构程序设计 .....            | 28 |



|                               |           |
|-------------------------------|-----------|
| 3.1.1 if 语句 .....             | 28        |
| 3.1.2 switch 语句 .....         | 33        |
| 3.2 循环结构程序设计 .....            | 35        |
| 3.2.1 while 语句 .....          | 35        |
| 3.2.2 do-while 语句 .....       | 36        |
| 3.2.3 for 语句 .....            | 37        |
| 3.2.4 循环嵌套 .....              | 38        |
| 3.3 break 和 continue 语句 ..... | 40        |
| 3.3.1 break 语句 .....          | 40        |
| 3.3.2 continue 语句 .....       | 41        |
| <b>第四章 类的定义</b> .....         | <b>43</b> |
| 4.1 面向对象思想的基本概念 .....         | 43        |
| 4.2 类的定义 .....                | 44        |
| 4.2.1 类的完整定义格式 .....          | 44        |
| 4.2.2 对象的创建、使用和清除 .....       | 47        |
| 4.2.3 使用对象作为方法的实际参数 .....     | 48        |
| 4.2.4 方法重载 .....              | 49        |
| 4.2.5 构造方法 .....              | 50        |
| 4.2.6 main() 方法 .....         | 52        |
| 4.3 类成员的修饰符 .....             | 52        |
| 4.3.1 访问控制 .....              | 52        |
| 4.3.2 static .....            | 54        |
| 4.4 类的继承 .....                | 56        |
| 4.4.1 继承 .....                | 56        |
| 4.4.2 成员变量的隐藏 .....           | 57        |
| 4.4.3 方法的覆盖 .....             | 58        |
| 4.4.4 抽象类 .....               | 59        |
| 4.4.5 多态 .....                | 61        |
| 4.4.6 final .....             | 62        |
| 4.5 接口与包 .....                | 63        |
| 4.5.1 接口的定义 .....             | 63        |
| 4.5.2 包的定义与使用 .....           | 65        |
| 4.5.3 类与类成员访问控制权限 .....       | 66        |
| 4.6 一个 Java 程序源文件的基本结构 .....  | 71        |
| <b>第五章 常用类与数组</b> .....       | <b>73</b> |
| 5.1 系统类库介绍 .....              | 73        |
| 5.1.1 Java 类库简介 .....         | 73        |
| 5.1.2 Java API 文档 .....       | 74        |
| 5.2 Object 类 .....            | 75        |





|       |                                                 |     |
|-------|-------------------------------------------------|-----|
| 5.3   | Math 类                                          | 76  |
| 5.4   | 基本类型的封装类                                        | 78  |
| 5.5   | 字符串处理                                           | 79  |
| 5.5.1 | String 类                                        | 79  |
| 5.5.2 | StringBuffer 类                                  | 81  |
| 5.5.3 | String 与 StringBuffer 使用“==”与 equals()方法进行比较的区别 | 84  |
| 5.5.4 | StringTokenizer 类                               | 86  |
| 5.6   | 数组                                              | 87  |
| 5.6.1 | 一维数组                                            | 87  |
| 5.6.2 | 多维数组                                            | 93  |
| 5.6.3 | 集合操作                                            | 95  |
| 第六章   | 异常处理                                            | 101 |
| 6.1   | 异常                                              | 101 |
| 6.1.1 | 基本概念                                            | 101 |
| 6.1.2 | Throwable 类及其子类                                 | 103 |
| 6.2   | 异常的种类                                           | 103 |
| 6.2.1 | 运行时的异常                                          | 103 |
| 6.2.2 | 捕获异常                                            | 105 |
| 6.2.3 | 声明异常                                            | 108 |
| 6.2.4 | 抛出异常                                            | 109 |
| 6.2.5 | 自定义异常                                           | 110 |
| 第七章   | 流操作                                             | 114 |
| 7.1   | 字节流                                             | 115 |
| 7.2   | 字符流                                             | 118 |
| 7.3   | 文件操作                                            | 119 |
| 7.3.1 | File 类与文件操作                                     | 119 |
| 7.3.2 | 随机存取文件流                                         | 123 |
| 7.4   | 网络通信                                            | 127 |
| 7.4.1 | URL                                             | 127 |
| 7.4.2 | Socket 通信                                       | 128 |
| 第八章   | 线程                                              | 135 |
| 8.1   | 线程的生命周期                                         | 135 |
| 8.1.1 | 多线程的概念                                          | 135 |
| 8.1.2 | 线程的生命周期                                         | 136 |
| 8.2   | 继承 Thread 类创建线程                                 | 138 |
| 8.3   | 实现 Runnable 接口创建线程                              | 139 |
| 8.4   | 线程的优先级                                          | 140 |
| 8.5   | 线程同步                                            | 142 |
| 8.6   | 在同步方法中使用 wait(), notify()和 notifyAll()方法        | 146 |



|                                  |            |
|----------------------------------|------------|
| <b>第九章 Applet 与 GUI 设计 .....</b> | <b>153</b> |
| 9.1 Applet .....                 | 153        |
| 9.1.1 Applet Java 小应用程序示例.....   | 153        |
| 9.1.2 Applet 运行原理 .....          | 155        |
| 9.2 事件委托模型 .....                 | 156        |
| 9.3 组件 .....                     | 161        |
| 9.3.1 AWT 包及其组件 .....            | 161        |
| 9.3.2 Swing 包及其组件 .....          | 168        |
| 9.4 布局管理器 .....                  | 173        |
| 9.5 综合实例 .....                   | 179        |



# 第一章 Java 初识

## 【学习目标】

了解 Java 语言的基本特点、发展过程以及应用前景;初步认识 Java 程序以及注释的用法;掌握 Java 开发环境配置的过程;理解 Java 虚拟机的平台无关性。

## 【知识要点】

Java 语言的特点、Java 程序基本结构、注释的用法、开发环境的配置。

编程的目的是让计算机按照程序员的要求来完成特定的工作。如同人与人之间需要用某种语言交谈一样,我们需要利用程序设计语言和计算机交流。人和计算机之间最原始的交流方式是采用机器语言,程序员用“0”和“1”告诉计算机该做什么,怎么做。但是这样编写程序非常烦琐且容易出错,因此人们不断地发展创造新的编程语言,提高编程效率。从机器语言到汇编语言,到高级语言(如 C 语言, Pascal 等),再到面向对象的语言(如 Java、C++ 等)。在编程语言不断的发展过程中,编程也就“相对”而言越来越简单了。

Java 是一门纯面向对象的程序设计语言,它自身也在不断的演化发展中。下面就让我们开始 Java 语言的学习。

## 1.1 Java 的起源与发展

### 1.1.1 Java 的起源

1991 年初,美国加州的 Sun Microsystem 公司(以下简称 Sun 公司)成立了一个以 James Gosling 为首、名为 Green 的项目研发小组,其目标是开发一个面向家用电器市场的软件产品,用软件实现一个对家用电器进行集成控制的小型控制装置。他们首先注意到这个产品必须具有平台独立性,即让该软件在任何 CPU 上都能运行。为达到此目的,Gosling 首先从改写 C++ 语言的编译器着手。但是,他们很快便意识到这个产品还必须具有高度的简洁性和安全性,而 C++ 在这方面显然无法胜任。因此,Gosling 决定自行开发一种新的语言,并将该语言命名为 Oak(橡树)。

Oak 是这个名为“\* 7”(StarSeven)产品中的一个组成部分。StarSeven 是一个集成了 Oak、GreenOS(一种操作系统)、用户接口模块和硬件模块四个部分的类似于 PDA(Personal Digital Assistant,个人数字助理)的设备。StarSeven 的第一个原型于 1992 年 8 月问世。尽管这个原型非常成功,但在竞争激烈的家用电器市场上却败给了竞争对手。

可怜的 Oak 几乎无家可归,就在举步维艰,随时会被取消的时刻,情况却发生了巨大变化。1993 年,因特网在美国开始大规模的发展,基于因特网的 WWW 也爆炸性地流行起来。



Sun 公司发现, Green Team 的项目研究成果——Oak (Java), 似乎天生就是为因特网而诞生的, 在得到了 Sun 公司首席执行官 Scott McNealy 的支持后, 触发了 Java 进军 Internet。

于是, Java 第一次以 Applet 的形式在因特网上出现了。Applet 不但使 WWW 页面显示静态的内容, 而且可以显示动态的内容和动画。同时实行了本地计算机, 从远程联网的服务器上下载资料并正确地显示出来。这些技术在当时引起了巨大的震撼, 促使 Java 在因特网上得以飞速地发展。1995 年 5 月 23 日, Sun 在 SunWorld'95 上正式发布 Java 和 HotJava 浏览器。而今, Java 已发展成为最流行的编程语言。

Java 的来历:

由于 Oak 这个名称与其他产品的名称相同, 所以开发小组要为这个语言取一个新名字。据说当时研发小组的成员经常在公司附近的一家咖啡厅喝咖啡, 而咖啡的原产地是 Java (爪哇岛), 于是有人就提议采用 Java 作为这门新语言的名字。

### 1.1.2 Java 语言的主要特点

#### 1. 简单

Java 与 C++ 很相似, 但却简单得多。它把一般高级编程语言中不是绝对需要的特性都删掉了。例如, Java 不用在程序中进行内存管理, 没有运算符重载、预处理、指针运算、结构、联合等。这样学习起来也就相对简单很多。

#### 2. 面向对象

Java 语言是纯面向对象的, 它不像 C++ 那样既支持面向对象的技术, 又支持面向过程的程序设计技术。在面向对象的技术中, 把现实世界中的任何事物, 都可以看做是对象。现实世界中对象所具有的属性 and 行为, 可以映射到计算机程序上。属性用数据表示, 行为用程序代码实现。面向对象的相关细节和概念, 在第四章内容中会有详细介绍。

#### 3. 可移植性(平台无关性)

程序的可移植性指的是程序不经修改而在不同硬件或软件平台上运行的特性。C 和 C++ 只具有一定程度的源代码级可移植性, 其源程序要想在不同平台上运行, 必须重新编译。而 Java 不仅源代码级是可移植的, 甚至源代码经过编译之后形成的二进制代码——字节码, 也同样是可移植的。Java 主要靠 Java 虚拟机(JVM)实现平台无关性。

#### 4. 安全性

C 和 C++ 程序在安全性和稳定性方面的最大问题在于其指针的使用, 不进行数组下标越界检查, 缺乏自动的内存管理等。使用指针的一个危险是它能够访问任意内存空间, 如果病毒利用指针进入操作系统的内存空间, 并在其中执行特权指令, 它就能随心所欲地进行破坏。而 Java 首先摒弃了指针数据类型, 这样, 程序员便不再能够凭借指针在任意内存空间中访问; 其次, Java 提供了数组下标越界检查机制, 从而避免溢出带来的问题; 第三, Java 提供了自动内存管理机制, 它可以利用系统的空闲时间来执行诸如必要的垃圾清除等操作。因此 Java 的安全性能大大提高了。

### 1.1.3 Java 语言的应用发展

Java 发展到今天, 已从编程语言发展成为全球第一大通用开发平台。Java 技术已被计算机行业主要公司所采纳。1999 年, Sun 公司推出了以 Java2 平台为核心的 J2EE、J2SE 和



J2ME 三大平台。随着三大平台的迅速推进,全球形成了一股巨大的 Java 应用浪潮。

### 1. Java 2 Platform, Micro Edition(J2ME)

Java 2 平台微型版,主要用于小型设备环境下的开发。Sun 公司将 J2ME 定义为“一种以广泛的消费性产品为目标、高度优化的 Java 运行环境”。通过 Java 的特性,遵循 J2ME 规范开发的 Java 程序可以运行在各种不同的小型设备上。现如今我们很多手机上都支持的 Java 游戏就是运用 Java 技术的典型代表。

### 2. Java 2 Platform, Standard Edition(J2SE)

Java 2 平台标准版,适用于桌面系统应用程序的开发。我们学习 Java 语言通常都在 J2SE 平台下进行,本书中的例程就是利用 J2SE 6.0 环境下编写通过的。

### 3. Java 2 Platform, Enterprise Edition(J2EE)

Java 2 平台企业版,主要利用 Java 2 平台来简化企业解决方案的开发、部署和管理等相关复杂问题的体系结构。J2EE 技术的核心就是 Java 2 平台的标准版,J2EE 不仅巩固了标准版的许多优点,例如:“一次编写、随处运行”的特性、方便存取数据库的 JDBC API、CORBA 技术,以及能够在 Internet 应用中保护数据的安全模式等,同时还提供了对 EJB(Enterprise Java Beans)、Java Servlets API、JSP(Java Server Pages),以及 XML 技术的全面支持。很多大型企业的网站或应用系统都是基于 J2EE 平台的,应用非常广泛。

## 1.2 Java 程序设计示例

### 1.2.1 Java 程序开发的环境配置

Sun 公司为 Java 程序员提供了一套免费的 Java 开发和运行环境,取名为 Java2 SDK(简称 JDK)。通常 JDK 指的就是 J2SE 开发环境,目前(2008 年 5 月)已经发展到第六版。

下载安装配置 JDK 开发环境的具体步骤如下:

1. 访问 Sun 公司的网站(<http://java.sun.com/javase/downloads/index.jsp>)下载最新的 Java 工具包,如图 1-1。(注意:Java 的开发帮助文档也可以在这里找到)由于我们通常都在 windows 系统环境下进行开发,所以我们选择 windows 版本的 JDK,如图 1-2。选择下载安装文件(jdk-6u6-windows-i586-p.exe),如图 1-3。

2. 双击运行下载文件 jdk-6u6-windows-i586-p.exe,程序会默认安装在“C:\Program Files\Java\jdk1.6.0\_06\”目录下。为了方便后面使用,本书假定安装在 D:\jdk6\目录中。通常在 JDK 安装目录下有 bin、demo、lib、jre 等子目录和一些文件,如图 1-4。其中 bin 目录保存了 javac、java、appletviewer 等命令文件;demo 目录保存了许多 Java 程序的例子;lib 目录保存了一些附加的类库文件;jre 目录保存的是 Java 的运行时环境等。具体说明可以查阅目录中的 README\_zh-CN.html 文件。

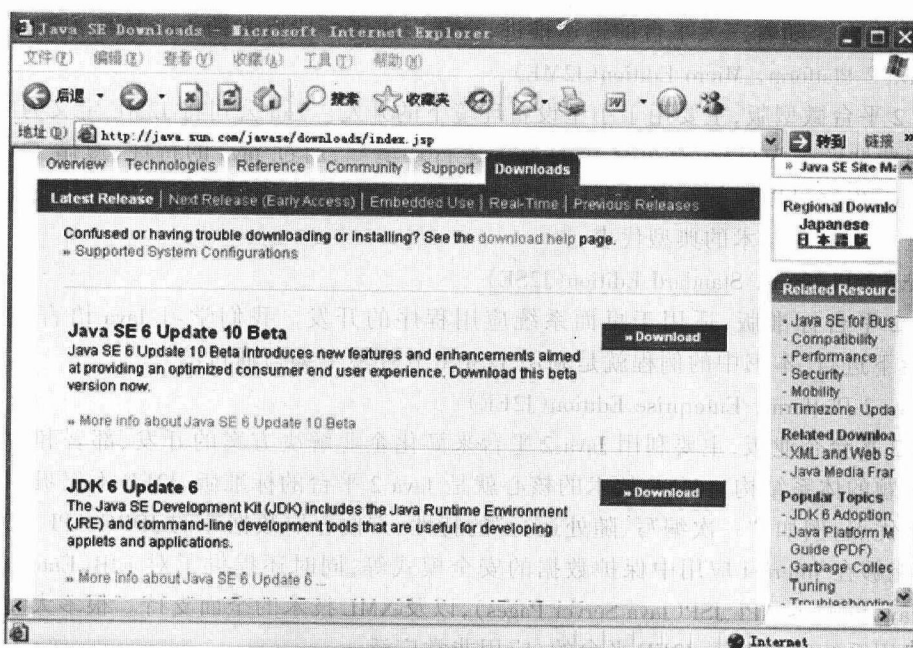


图 1-1 选择最新版本 JDK 6 update 6

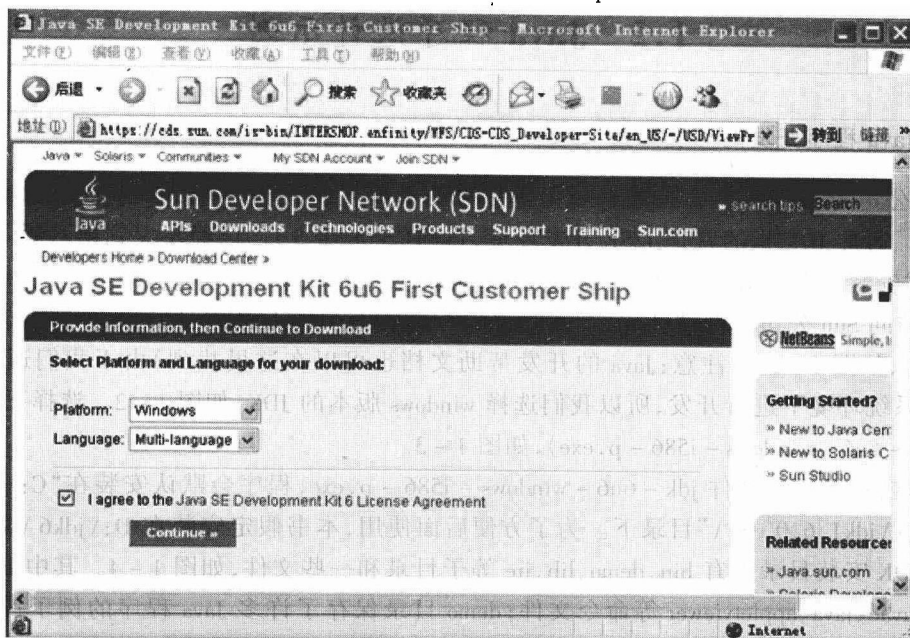


图 1-2 选择 windows 平台下的 JDK



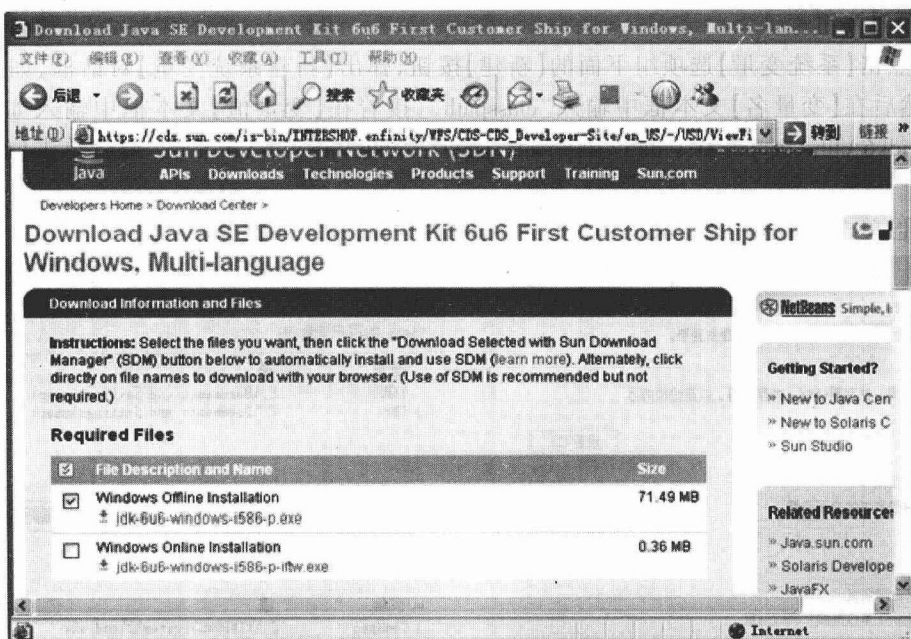


图 1-3 选择下载 jdk-6u6-windows-i586-p.exe



图 1-4 Java 目录基本结构

3. 在安装完 JDK 之后,必须配置系统命令路径环境变量 path 和类路径 classpath, JDK 才能够正常使用。在 windows XP 操作系统中设置方法如下:

- (1) 右键单击桌面上的【我的电脑】,选择【属性】菜单,弹出【系统属性】对话框,如图 1-5;
- (2) 选择【高级】选项卡中的【环境变量】按钮,弹出【环境变量】对话框,如图 1-6;
- (3) 在【系统变量】选项组里找到已经存在的“path”变量,单击【编辑】按钮弹出【编辑系统变量】对话框。在变量值后面接着原有内容追加“;.\D:\jdk6\bin\”,如图 1-7,,然后



单击【确定】按钮；

(4) 点击【系统变量】选项组下面的【新建】按钮，弹出【新建系统变量】对话框，如图 1-8 所示。然后在【变量名】文本框中输入“classpath”，接着在【变量值】文本框中输入如下内容“.;D:\jdk6\lib\dt.jar;D:\jdk6\lib\tools.jar;D:\jdk6\jre\lib\rt.jar”，最后单击【确定】按钮；

(5) 再单击【确定】按钮，关闭【环境变量】对话框和【系统属性】对话框。

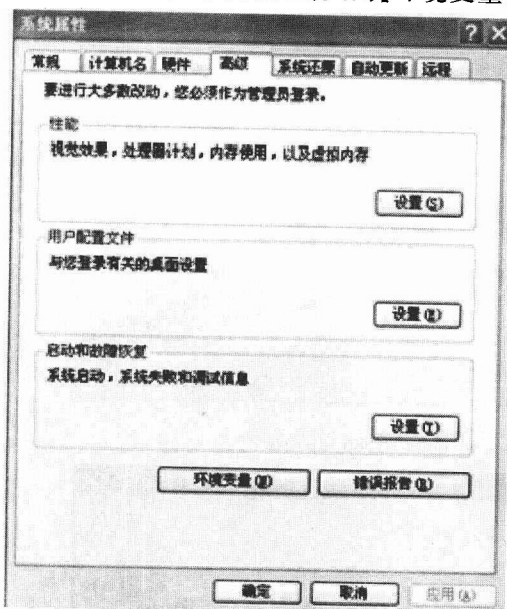


图 1-5 【系统属性】对话框

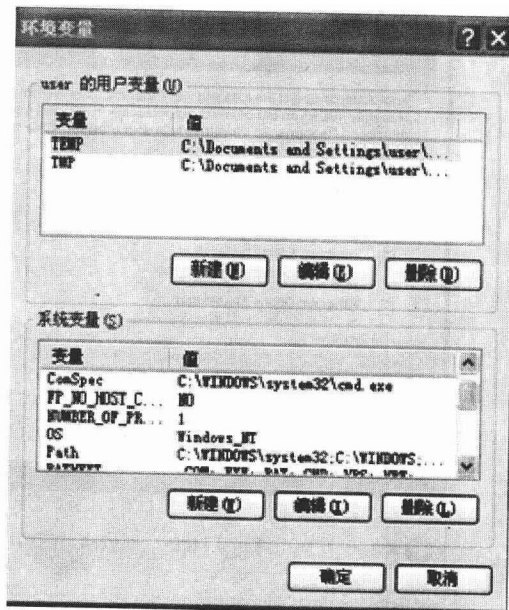


图 1-6 【环境变量】对话框

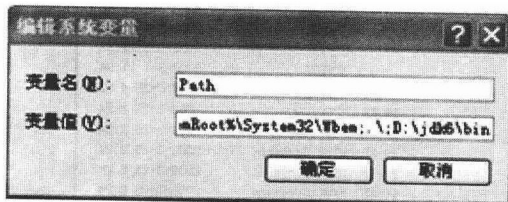


图 1-7 修改 path 路径

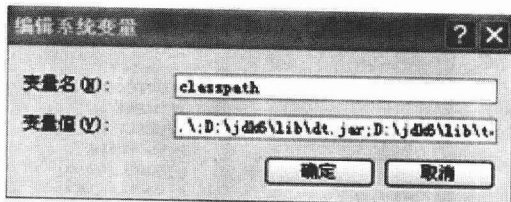


图 1-8 增加 classpath 路径

系统路径(path)的作用是在当在 Dos 命令行窗口下执行命令时，系统能正确地自动找到命令文件所在的位置。类路径(classpath)的作用是，当系统在编译 Java 文件时能正确地自动找到源文件中所引用类库中的类文件。上面配置中的目录“D:\jdk6”是根据前面具体安装的情况决定的，如果你的安装目录和书上的不一致，那么你要按照自己的安装目录来设置，不可盲目照抄。“.”表示的是当前目录，它的作用是使得我们不管进入哪一个目录，系统都能方便地找到这个目录，从而访问此目录下面的文件。设置系统变量时，所增加的内容之间都是用“;”号分隔，注意不要输错。我们在 classpath 中添加的 jar 文件其实就是类库，本身 jar 文件是一个压缩包，里面放有很多类文件。除了在 jdk 安装目录下的 lib 目录和 jre\lib\目录的 jar 文件以外，如果想用到其他第三方的包或者是我们自己写的类库也可以把它们的路径用“;”分隔加在 classpath 后面。配置好后，点击【开始】→【运行】，在弹出的【运行】对话框

中输入“cmd”，点击【确定】后弹出 Dos 窗口。接着我们输入“javac”命令，如果出现相关用法参数的说明信息，那么说明系统变量 path 配置正确，如图 1-9。值得注意的是，如果在打开 Dos 窗口后修改了前面的配置，那么必须关掉当前 Dos 窗口再重新打开，新配置的环境变量才能生效。

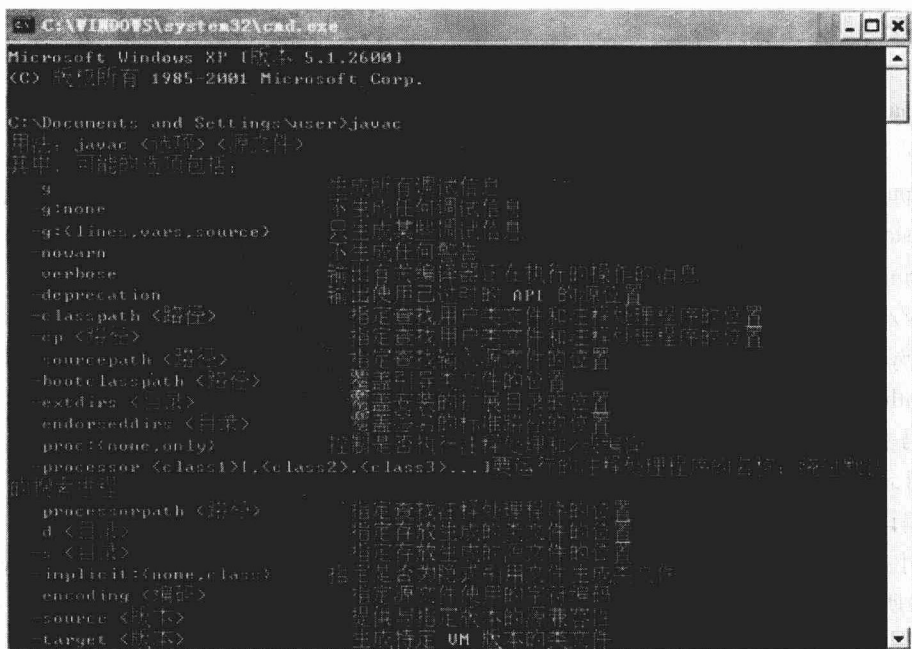


图 1-9 在 Dos 窗口下运行 javac 命令

现在我们可以利用文本编辑器来编写 Java 程序了。例如用 windows 自带的“记事本”或者是功能比较强大的“ultraedit”等文本编辑器均可。除了利用文本编辑器来编写代码外，市面上还有一些常见的集成开发工具，如 Eclipse、NetBeans、JBuilder 等，它们分别是 IBM、Sun、Borland 公司支持的产品。它们不但集成了 JDK 和文本编辑器，还提供了很多有用的工具，非常有利于团队或个人的快速开发。但对于我们初学者来说，一般还是先用文本编辑器编写代码较好，因为这样有利于熟悉 Java 代码。等到对于 Java 熟练掌握到一定程度后，再学习利用集成开发环境来编写程序。

### 1.2.2 编写第一个 Java 应用程序

编写 Java 应用程序，通常按照如下步骤进行：

1. 打开文本编辑器，输入如下的程序代码：

【例 1-1】

/\*

这是我们编写的第一个 Java 应用程序，我们通过它来初步掌握一个 Java 程序应该如何编写。

\*/

```
public class FirstJavaDemo{
    public static void main(String args[ ]){
```