



软件职业技术学院“十一五”规划教材

Java

程序设计技能教程

主 编 彭德林 李德有 副主编 柴方艳 刘明刚 李继连
主 审 解晨光 钱英军



中国水利水电出版社
www.waterpub.com.cn

软件职业技术学院“十一五”规划教材

Java 程序设计技能教程

主 编 彭德林 李德有

副主编 柴方艳 刘明刚 李继连

主 审 解晨光 钱英军

内 容 提 要

Java 语言作为网络时代程序设计的宠儿,越来越受到人们的青睐,现在已经成为最重要的软件开发工具之一。对于计算机专业学生来说,Java 是必须具备的技能之一。虽然现在关于 Java 的教材很多,但与社会实际应用相接轨的教材却不多,本书以社会实际需求为主线,采用应用得最广的 Eclipse 开发平台,采用任务驱动模式进行内容编排,使学生真正能学有所用,并且配备与教材内容相关的实验指导书,更有利于教师的实验教学。

本教材在强调使学生全面掌握 Java 基本操作的基础上,把知识点与应用案例相结合,使学生学习起来有的放矢,操作时也更加得心应手。本书主要内容包括 Java 语言基础、Java 面向对象程序设计等,在“任务驱动式教学”思想的指导下,每章都设置了若干个任务,都以相关应用操作为主线进行叙述,通过案例进行应用剖析,相关章节配有实训操作和习题,并在书后提供习题答案。

本书深入浅出,操作性强,以实际应用为切入点,符合高职高专教学特点,适合高职高专学生使用,也可作为各类 Java 技能培训班的教材。

本书配有电子教案及相关教学资源,读者可以到中国水利水电出版社和万水书苑的网站上免费下载,网址为: <http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

图书在版编目(CIP)数据

Java 程序设计技能教程 / 彭德林, 李德有主编. —北京:

中国水利水电出版社, 2009

软件职业技术学院“十一五”规划教材

ISBN 978-7-5084-6610-1

I. J… II. ①彭…②李… III. JAVA 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 109884 号

策划编辑: 石永峰 责任编辑: 李 炎 加工编辑: 徐 雯 封面设计: 李 佳

书 名	软件职业技术学院“十一五”规划教材 Java 程序设计技能教程
作 者	主 编 彭德林 李德有 副主编 柴方艳 刘明刚 李继连 主 审 解晨光 钱英军
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	184mm×260mm 16 开本 17.5 印张 432 千字
版 次	2009 年 7 月第 1 版 2009 年 7 月第 1 次印刷
印 数	0001—4000 册
定 价	28.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

序

随着信息技术的广泛应用和互联网的迅猛发展,以信息产业发展水平为主要特征的综合国力竞争日趋激烈,软件产业作为信息产业的核心和国民经济信息化的基础,越来越受到世界各国的高度重视。中国加入世贸组织后,必须以积极的姿态,在更大范围和更深程度上参与国际合作和竞争。在这种形势下,摆在我们面前的突出问题是人才短缺,计算机应用与软件技术专业领域技能型人才的缺乏尤为突出,无论是数量还是质量,都远不能适应国内软件产业的发展和信息化建设的需要。因此,深化教育教学改革,推动高等职业教育与培训的全面发展,大力提高教学质量,是迫在眉睫的重要任务。

2000年6月,国务院发布《鼓励软件产业和集成电路产业发展的若干政策》,明确提出鼓励资金、人才等资源投向软件产业,并要求教育部门根据市场需求进一步扩大软件人才培养规模,依托高等学校、科研院所,建立一批软件人才培养基地。2002年9月,国务院办公厅转发了国务院信息化工作办公室制定的《振兴软件产业行动纲要》,该《纲要》明确提出要改善软件人才结构,大规模培养软件初级编程人员,满足软件工业化生产的需要。教育部也于2001年12月在35所大学启动了示范性软件学院的建设工作,并于2003年11月启动了试办示范性软件职业技术学院的建设工作。

示范性软件职业技术学院的建设目标是:经过几年努力,建设一批能够培养大量具有竞争能力的实用型软件职业技术人才的基地,面向就业、产学结合,为我国专科层次软件职业技术人才培养起到示范作用,并以此推动高等职业技术教育人才培养体系与管理体制和运行机制的改革。要达到这个目标,建立一套适合软件职业技术学院人才培养模式的教材体系显得尤为重要。

高职高专的教材建设已经走过了几个发展阶段,由最开始本科教材的压缩到加大实践教学环节的比重,再到强调实践性教学环节,但是学生在学习时还是反映存在理论与实践的结合问题。为此,中国水利水电出版社在经过深入调查研究后,组织了一批长期工作在高职高专教学一线的老师,编写了这套“软件职业技术学院‘十一五’规划教材”,本套教材采用项目驱动的方法来编写,即全书所有章节都以实例作引导来说明各知识点,各章实例之间并不是孤立的,每个实例都可以作为最终项目的一个组成部分;每一章章末还配有实习实训(或叫实验),这些实训组合起来是一个完整的项目。

采用这种方式编写的图书与市场上同类教材相比更具优越性,学生不仅仅学到了知识点,还通过项目将这些知识点连成一条线,开拓了思路,掌握了知识,达到了面向岗位的职业教育培训目标。

本套教材的主要特点有:

- (1) 课程主辅分明——重点突出,教学内容实用。
- (2) 内容衔接合理——完全按项目运作所需的知识体系结构设置。

(3) 突出实习实训——重在培养学生的专业能力和实践能力,力求缩短人才与企业间的磨合期。

(4) 教材配套齐全——本套教材不仅包括教学用书,还包括实习实训材料、教学课件等,使用方便。

本套教材适用于广大计算机专业和非计算机专业的大中专院校的学生学习,也可作为有志于学习计算机软件技术与开发的工程技术人员的参考教材。

编委会

2006年7月

前 言

Java 语言经过最近十多年的快速发展, 技术越来越成熟, 应用也越来越广, 在软件开发方面的霸主地位更加稳固, 也已经是计算机类学生必修的课程之一。教材的优劣直接影响教学效果的好坏, 本书在保证知识体系完整性的前提下, 采用任务驱动式教学方法, 以社会上应用得最广的 Eclipse 为开发平台, 突出实用性、针对性, 引导学生“在做中学”, 培养学生分析问题解决问题的能力, 把提高学生动手能力和综合素质作为首要任务。

全书共分 11 章, 第 1 章简介 Java 软件开发相关技术; 第 2 章介绍 Eclipse 基本操作; 第 3 章介绍 Java 基础; 第 4 章介绍基本控制结构; 第 5 章介绍类和对象; 第 6 章介绍类的继承与多态; 第 7 章介绍异常处理和多线程; 第 8 章介绍输入输出流; 第 9 章介绍图形用户界面设计; 第 10 章介绍键盘与鼠标事件的响应; 第 11 章介绍数据库编程。每章后面都配有实训以提高学生的动手能力, 并配有“练习与提高”环节以检验学生对知识的掌握程度, 还提供了相应的答案。

本书内容安排合理, 逻辑性强, 讲解循序渐进, 通俗易懂, 既适合高职高专计算机及相关专业学生作为教材使用, 也适合各种 Java 语言培训机构使用。

本书由彭德林、李德有任主编, 柴方艳、刘明刚、李继连任副主编, 解晨光、钱英军任主审。全书由彭德林、李德有审阅定稿。其中, 第 1 章由李继连编写, 第 2 章由李德有编写, 第 3 章由刘妍编写, 第 4 章由郭瑞波编写, 第 5 章由毛敏芹编写, 第 6 章由刘明刚编写, 第 7 章由相成编写, 第 8 章由陈员义编写, 第 9 章由孙守梅编写, 第 10 章由赵玉兰编写, 第 11 章由柴方艳编写, 第 12 章由彭德林编写。解晨光对全文进行了审核校对。

本书在编写过程中得到了中国水利水电出版社有关领导和编辑的大力支持和帮助, 在此一并表示感谢。由于编者水平有限, 书中难免出现错误和不足, 敬请广大读者和同仁给予批评指正。

作 者
2009 年 4 月

目 录

序

前言

第1章 Java 软件开发相关技术简介.....1	2.3.1 Eclipse 结构体系.....22
本章目的及任务.....1	2.3.2 Eclipse 工作台.....23
1.1 任务一 掌握 Java 语言基础知识.....1	2.3.3 Eclipse 资源管理.....26
1.1.1 Java 语言的发展.....1	2.4 任务四 使用 Eclipse 进行 Java 编程.....27
1.1.2 Java 程序开发与运行环境.....2	2.4.1 编辑 Java 程序.....27
1.1.3 Java 语言特点.....3	2.4.2 编译和运行程序.....29
1.1.4 Java 程序分类.....4	2.4.3 调试程序.....30
1.1.5 JDK 的安装与环境变量配置.....4	2.5 任务五 实训二 Eclipse 的安装配置
1.1.6 Java Application 程序结构及运行过程.....7	及 Java 程序的调试运行.....31
1.2 任务二 掌握 Java 语言体系.....8	本章小结.....33
1.2.1 J2SE 概述.....8	练习与提高二.....33
1.2.2 J2EE 概述.....9	第3章 Java 基础.....35
1.2.3 J2ME 概述.....13	本章目的及任务.....35
1.3 任务三 了解 Java 开发工具.....13	3.1 任务一 掌握 Java 关键字、标识符.....35
1.3.1 JBuilder 简介.....13	3.1.1 Java 语言的关键字.....35
1.3.2 Eclipse 简介.....14	3.1.2 Java 语言标识符和命名约定.....36
1.3.3 其他开发工具简介.....15	3.1.3 Java 程序中的分隔符.....36
1.4 任务四 实训一 JDK 安装配置及 Java	3.2 任务二 掌握 Java 语言的基本数据类
程序的编译运行.....16	型、常量和变量.....37
本章小结.....17	3.2.1 基本数据类型.....37
练习与提高一.....17	3.2.2 常量.....38
第2章 Eclipse 基本操作.....19	3.2.3 变量的定义及作用范围.....38
本章目的及任务.....19	3.2.4 基本数据类型间的转换.....40
2.1 任务一 了解 Eclipse 概况.....19	3.3 任务三 掌握 Java 运算符和表达式.....41
2.1.1 Eclipse 的历史与发展.....19	3.3.1 运算符.....41
2.1.2 Eclipse 的特点.....19	3.3.2 表达式.....48
2.2 任务二 掌握 Eclipse 的安装与配置.....20	3.4 任务四 掌握 Java 数组及字符串的
2.2.1 Eclipse 下载与安装.....20	使用方法.....48
2.2.2 Eclipse 的多国语言包的安装.....20	3.4.1 数组的声明与创建.....48
2.2.3 Eclipse 相关配置.....21	3.4.2 数组的初始化及数组元素的使用.....48
2.3 任务三 熟悉 Eclipse 工作环境.....22	3.4.3 字符数组与字符串 String 类.....50

3.4.4 字符串数组.....	51	本章目的及任务.....	104
3.4.5 字符串相关操作.....	51	6.1 任务一 掌握类的继承.....	104
3.5 任务五 实训三 数组及字符串的使用.....	56	6.1.1 继承的基本概念.....	104
3.5.1 数组的创建及使用.....	56	6.1.2 继承的实现.....	107
3.5.2 字符串的使用.....	57	6.1.3 super 与 this 的使用.....	108
本章小结.....	57	6.1.4 final 类和 final 方法.....	112
练习与提高三.....	57	6.2 任务二 掌握类的多态.....	113
第4章 基本控制结构	60	6.2.1 多态性的含义.....	113
本章目的及任务.....	60	6.2.2 方法的重载.....	113
4.1 任务一 掌握 Java 分支结构.....	60	6.2.3 方法的覆盖.....	117
4.1.1 if 语句.....	60	6.3 任务三 掌握接口的使用.....	121
4.1.2 switch 语句.....	64	6.3.1 抽象类和抽象方法.....	121
4.2 任务二 掌握 Java 循环结构.....	66	6.3.2 接口的声明.....	122
4.2.1 for 语句.....	67	6.3.3 接口的使用.....	122
4.2.2 do-while 语句.....	69	6.4 任务四 掌握包的使用.....	123
4.2.3 while 语句.....	71	6.4.1 包的概念.....	123
4.2.4 嵌套循环.....	72	6.4.2 包的使用.....	124
4.3 任务三 实训四 分支及循环结构编程.....	74	6.4.3 常用系统包.....	125
本章小结.....	75	6.5 任务五 实训六 继承与多态编程实训.....	125
练习与提高四.....	76	练习与提高六.....	126
第5章 类和对象	78	第7章 异常处理和多线程	129
本章目的及任务.....	78	本章的目的及任务.....	129
5.1 任务一 掌握类和对象的基本知识.....	78	7.1 任务一 掌握 Java 中对异常的处理.....	129
5.1.1 面向对象的基本概念.....	78	7.1.1 异常类和异常处理机制.....	129
5.1.2 类的声明.....	80	7.1.2 程序中异常处理方法.....	131
5.1.3 方法的声明.....	83	7.2 任务二 掌握程序对多线程的处理.....	138
5.1.4 方法调用及参数传递.....	84	7.2.1 多线程的概念.....	138
5.1.5 对象的创建和使用.....	87	7.2.2 实现多线程的两种方法.....	138
5.1.6 构造方法及特点.....	90	7.2.3 线程的生命周期和控制.....	139
5.1.7 对象的初始化.....	93	7.2.4 线程的同步.....	143
5.2 任务二 掌握类的封装.....	94	7.3 任务三 实训七 异常处理与多线程编程实训.....	146
5.2.1 访问权限.....	94	本章小结.....	149
5.2.2 类变量和实例变量.....	96	练习与提高七.....	149
5.2.3 类成员方法和函数数学方法.....	98	第8章 输入输出流	152
5.3 任务三 实训五 类和对象.....	100	本章目的及任务.....	152
本章小结.....	101	8.1 任务一 掌握流及标准输入输出.....	152
练习与提高五.....	101	8.1.1 流及分类.....	152
第6章 类的继承与多态	104		

8.1.2 标准输入	155	9.3.3 字体与颜色的使用	196
8.1.3 标准输出和标准错误	156	9.3.4 按钮	198
8.2 任务二 掌握常用输入输出流的使用		9.3.5 文本框、文本区和标签	199
方法	158	9.3.6 复选框与单选按钮	201
8.2.1 字节流 InputStream 和 OutputStream		9.3.7 列表框和组合框	205
类简介	158	9.3.8 对话框和文件选择对话框	208
8.2.2 字符流 Reader 和 Writer 类简介	160	9.3.9 菜单	214
8.2.3 FileInputStream 类的使用	161	9.4 任务四 掌握图形的绘制方法	218
8.2.4 FileOutputStream 类的使用	163	9.4.1 坐标系统	218
8.2.5 FileReader 类和 FileWriter 类的		9.4.2 显示字符串	219
使用	164	9.4.3 绘制直线与矩形	219
8.2.6 使用 BufferedReader 类实现各种		9.4.4 绘制圆、椭圆与圆弧	220
数据类型的输入	167	9.4.5 绘制多边形与带颜色图形	220
8.3 任务三 掌握常用目录和文件操作	170	9.5 任务五 实训九 图形界面编程	
8.3.1 获取文件信息和更改文件属性	170	综合实训	220
8.3.2 列目录文件清单	173	本章小结	221
8.3.3 获取根目录和创建子目录	173	练习与提高九	221
8.3.4 文件的创建、更名和删除	174	第 10 章 键盘与鼠标事件的响应	224
8.3.5 基本数据类型数据的读写	174	本章目的及任务	224
8.3.6 使用增加了缓冲的流类	175	10.1 任务一 掌握事件处理机制	224
8.3.7 读取文件指定行	175	10.1.1 相关概念	224
8.4 任务四 实训八 文件输入输出操作	176	10.1.2 事件处理机制	224
本章小结	177	10.2 任务二 掌握对键盘事件响应的方法	227
练习与提高八	177	10.2.1 键盘事件	227
第 9 章 图形用户界面设计	180	10.2.2 键盘监听接口 KeyListener	228
本章目的及任务	180	10.2.3 键盘适配器类 KeyAdapter	229
9.1 任务一 了解与图形处理相关的包	181	10.3 任务三 掌握对鼠标事件的响应方法	230
9.1.1 java.awt 包	181	10.3.1 鼠标事件	230
9.1.2 java.swing 包	182	10.3.2 使用 MouseListener 接口处理	
9.1.3 java.awt.event 包	184	鼠标事件	231
9.2 任务三 掌握 Java Applet 的程序结构		10.3.3 使用 MouseMotionListener 接口	
及运行过程	184	处理鼠标事件	232
9.2.1 Applet 类及相关方法	185	10.4 任务四 实训十 键盘与鼠标事件	
9.2.2 Applet 程序建立及运行过程	187	响应实训	234
9.3 任务三 掌握使用 Swing 组件进行常		本章小结	235
用图形界面设计的方法	188	练习与提高十	235
9.3.1 窗口容器类 JFrame 使用	189	第 11 章 数据库编程	237
9.3.2 容器的布局	190	本章目的及任务	237

11.1 任务一 使用 JDBC 连接数据库	237	11.3.1 数据库编程过程	244
11.1.1 JDBC 简介	237	11.3.2 数据库编程实例	250
11.1.2 JDBC 驱动	239	11.4 任务四 实训十一 数据库编程实训 ..	256
11.2 任务二 建立数据源	240	本章小结	257
11.2.1 建立 Access 数据库	240	练习与提高十一	257
11.2.2 建立数据源	242	练习与提高参考答案	258
11.3 任务三 数据库编程	244	参考文献	270

第 1 章 Java 软件开发相关技术简介

本章目的及任务

本章目的

- 了解 Java 开发工具
- 掌握 Java 语言基本情况
- 掌握 Java 语言体系
- 掌握 JDK 安装配置及 Java 程序的编译运行

本章任务

本章包含如下四个任务：

- 任务一 掌握 Java 语言基础知识
- 任务二 掌握 Java 语言体系
- 任务三 了解 Java 开发工具
- 任务四 实训一 JDK 安装配置及 Java 程序的编译运行

1.1 任务一 掌握 Java 语言基础知识

1.1.1 Java 语言的发展

Java 是由 Sun 公司开发的新一代编程语言。使用它可在不同计算机、不同操作平台的网络环境中开发软件。不论你使用的是哪一种 WWW 浏览器，哪一种计算机，哪一种操作系统，只要 WWW 浏览器支持 Java，就可以看到生动的页面。Java 正在逐步成为 Internet 应用的主要开发语言，它彻底改变了应用软件的开发模式，带来了自个人计算机以来又一次技术革命，为迅速发展的信息世界增添了新的活力。

Sun 的 Java 语言开发小组成立于 1991 年，其目的是开拓消费类电子产品市场，例如交互式电视、烤面包箱等。1994 年，WWW 已如火如荼地发展起来。开发小组意识到 WWW 需要一个中性的浏览器，它不依赖于任何硬件平台和软件平台，是一种实时性较高、可靠安全、有交互功能的浏览器。于是，开发小组决定用 Java 开发一个新的 Web 浏览器。Java 1.0 版终于在 1996 年年初正式发表。

Java 虽出现的时间不长，但已被业界接受，IBM、Apple、DEC、Adobe、Silicon Graphics、HP、Oracle、Toshiba、Netscape 和 Microsoft 等大公司已经购买了 Java 的许可证。Microsoft 还在其 Web 浏览器 Explorer 3.0 版中增加了对 Java 的支持。

另外,众多的软件开发商也开发了许多支持 Java 的软件产品。如 Borland 公司的基于 Java 的快速应用程序开发环境 Latte; Metrowerks 公司和 Natural Intelligence 公司分别开发的基于 Macintosh 的 Java 开发工具; Sun 公司的 Java 开发环境 Java Workshop; Microsoft 也开发出了系列 Java 产品。数据库厂商如 Sybase、Versant、Oracle 都在开发支持 HTML 和 Java 的 CGI (Common Gateway Interface)。在以网络为中心的计算时代,不支持 HTML 和 Java,就意味着应用程序的应用范围只能限于同质的环境。

Java 的出现是计算机信息交换的一个重要里程碑。在单机时代,程序进程之间靠共享存储进行变量交换;在网络时代,运行在不同宿主机上的程序按网络协议进行无格式的消息(二进制字符流)交换,消息的语义由交换程序双方维护;在 Java 时代,运行在网络上的程序进程交换的是小应用程序(Applet)。小应用程序是一个对象,由一组属性和方法构成,是一个可执行的实体;不仅有数据的状态,而且有定义在数据上的操作。未来可能进行代表(Agent)交换,代表交换有一定的智能性,是信息交换的更高级阶段。

1.1.2 Java 程序开发与运行环境

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 开发工具集(Java Developers Kits, JDK)。编程人员和最终用户可以利用这些工具来开发 Java 程序或调用 Java 内容。

JDK 包括以下工具:

- javac: Java 语言编译器,输出结果为 Java 字节码。
- java: Java 字节码解释器。
- javap Disassembler: Java 字节码分解程序,本程序返回 Java 程序的成员变量及方法等信息。
- javaprof: 资源分析工具,用于分析 Java 程序在运行过程中调用了哪些资源,包括类和方法的调用次数和时间,以及各数据类型的内存使用情况等。
- javah: C 代码处理工具,用于从 Java 类调用 C++代码。
- java Applet Viewer: 小应用程序浏览工具,用于测试并运行 Java 小应用程序。
- java Debugger API: Java 调试工具。
- API Prototype Debugger: Java 调试工具原型。

(1) Java 编译器。

Java 程序的编译程序是 javac.exe。javac 命令将 Java 程序编译成字节码,然后用户可用 Java 解释器 Java 命令来解释执行这些 Java 字节码。Java 程序源码必须存放在后缀为.java 的文件里。对于 Java 程序里的每一个类,javac 都将生成与类相同名称但后缀为.class 文件。编译器把.class 文件放在.java 文件的同一个目录里,除非用户使用了-d 选项。

(2) Java 调试器。

JDB (Java Debugger) 是 J2SDK 提供的一个调试工具,可以实现单步跟踪、断点设置、监视程序输出情况等功能。在早期 Beta 1 版的 Java 调试器中,JDB 是命令行形式的,如 Sun 公司的 DBX 调试器。用 JDB 来调试 Java 应用程序,在调试之前,要确定应用程序是带标志-g 编译的,例如 javac -g HelloWorld.java。

(3) Java 解释器。

Java 命令是由 Java 编译器 `javac` 输出的 Java 字节码。`classname` 参数是要执行的类名称。注意任意在类名称后的参数都将传递给要执行类的 `main` 函数。Java 执行完 `main` 函数后退出，除非 `main` 函数创建了一个或多个线程。如果 `main` 函数创建其他线程，Java 总是等到最后一个线程退出才退出。

1.1.3 Java 语言特点

Java 语言具有能独立于平台而运行、面向对象、可对动态画面进行设计与操作、坚固等特点，又具有多线程、内置校验器以防止病毒入侵等功能，所以用于在 Internet 上研制与开发软件时，特别受到用户的欢迎。

1. 简单

由于 Java 的结构类似于 C 和 C++，所以一般熟悉 C 与 C++ 语言的编程人员稍加学习就不难掌握 Java 的编程技术了。并且 Java 所具有的自动内存管理机制也大大简化了 Java 程序设计开发过程。

2. 面向对象

简单地说，面向对象设计就是一种以数据（对象）以及其接口为重心的程序设计技术。面向对象的设计可以说是定义程序模块如何“即插即用”的机制。Java 的面向对象机制实际上可以看作是 C++ 面向对象机制的延伸。Java 提供了简单的类机制和动态的构架模型，对象中封装了它的状态变量和方法（函数、过程），实现了模块化和信息隐藏；而类则提供了一类对象的原型，通过继承和重载机制，子类可以使用或重新定义父类或者超类所提供的过程，从而实现代码的复用。

3. 自动内存管理

Java 的自动无用内存回收（auto garbage collection）实现了内存的自动管理，因此简化了 Java 程序开发的工作，早期的 gc（garbage collection）对系统资源抢占太多而影响整个系统的运行，Java2 对 gc 进行的改良使 Java 的效率有了很大提高。gc 的工作机制是周期性的自动回收无用存储单元。Java 的自动内存回收机制简化程序开发的同时，提高了程序的稳定性和可靠性。

4. 分布计算

Java 为程序开发提供了 `Java.net` 包，该包提供了一组使程序开发者可以轻易实现基于 TCP/IP 的分布式应用系统。此外，Java 还提供了专门针对互联网应用的类库，如 `URL`、`Java mail` 等。

5. 稳定性

人们最常见的应用程序错误就是“非法访问 xxx 内存”，其实质是程序指针使用出错。Java 拥有一种指针（pointer）模型，能够排除内存被覆盖和毁损数据的可能性。Java 不采用指针算术法，而是提供真正的数组（Array），运行程序下标检查；另外，它也不会发生将一个任意数转换成指针的情形。

6. 安全性

Java 的设计目的是提供一个用于网络/分布式的计算环境。因此，Java 强调安全性，如确保无病毒、小应用程序运行安全控制等。Java 的验证技术以公钥（public-key）加密算法为基础，而且从环境变量、类加载器、文件系统、网络资源和名字空间等方面实施安全策略。

7. 解释执行

Java 解释器 (interpreter) 可以直接在任何已移植的解释器的机器上解释、执行 Java 字节码, 不需重新编译。当然, 其版本向上兼容, 因此如果是高版本环境下编译的 Java 字节码, 在低版本环境下运行也许会出现问题。

8. 跨异构环境

Java 是网络空间的“世界语”, 编译后的 Java 字节码是一种结构中立性 (architecture neutral) 的目标文件格式, 可以在所有提供 Java 虚拟机 (JVM) 的多种不同主机、不同处理器上运行。

9. 平滑移植

“write once, run every where!” 也许是 Java 最诱人的特点。用 Java 开发而成的系统, 其移植工作几乎为零, 一般情况下只需对配置文件、批处理文件作相应修改即可实现平滑移植。

10. 多线程

Java 的多线程 (multithreading) 机制使程序可以并行运行。Java 还有一组同步化基本单元, 它们是以广泛使用的 C.A.R.Hoare 监视器与条件变量图为基础的。同步机制保证了对共享数据的正确操作。多线程使程序设计者可以用不同的线程分别实现各种不同的行为, 而不需要采用全局的事件循环机制, 因此使用 Java 语言可以非常轻松地实现网络上的实时交互行为。

11. 异常处理

C 语言程序员大都使用 goto 语句来进行条件跳转, 但 Java 编程不支持 goto 语句。Java 采用异常模型使程序的主流逻辑变得更加清晰明了, 并且能够简化错误处理工作。

12. 可扩充

Java 目前发布的 J2EE 标准主要为采用 Java 技术为企业提供全面解决方案提供了一个技术规范框架, 规划了一个利用现有和未来各种 Java 技术整合解决企业应用的远景蓝图。

1.1.4 Java 程序分类

Java 程序分为两种类型: Java 应用程序 (Java Application) 和 Java 小应用程序 (Java Applet)。Java Application 可以独立运行; Java Applet 不能独立运行, 但可以使用 Appletviewer 或其他支持 Java 的浏览器运行。无论哪种 Java 程序, 都用扩展名为 .java 的文件保存。

Application 和 Applet 是两个概念, 前者指的是一个完整程序, 后者指的是一个小程序, 作为一个完整程序具有一定的独立性, 玩过手机游戏的人都知道一个手机游戏对应一个 jar 文件, 这个 jar 文件就相当于一个 Application, 只需要选中它运行即可游戏, 不需要先启动另外一个程序后才能启动它。而 Applet 则需要先启动浏览器后才能运行它。由于较小, Applet 可以被看作一个软件组件, 它可以作为一个动态网站的一个组成部分, Applet 的主要用途是完成复杂的逻辑, 它在 JavaScript 等脚本语言和 JSP、ASP 等动态网页技术还没出现时负责动态地显示数据, 类似微软的 ActiveX 组件。

1.1.5 JDK 的安装与环境变量配置

JDK (Java Developer's Kit) 即 Java 开发者工具包, 也称为 J2SDK (Java 2 Software Development Kit), 是 SUN 公司提供的基础 Java 语言开发工具, 该工具软件包含 Java 语言的编译工具、运行工具以及执行程序的环境 (即 JRE)。JDK 现在是一个开源、免费的工具。JDK 是其他 Java 开发工具的基础, 也就是说, 在安装其他开发工具以前, 必须首先安装 JDK。对

于初学者来说,使用该开发工具进行学习,可以在学习的初期把精力放在对 Java 语言语法的学习上,体会更多底层的知识,对于以后的程序开发很有帮助。

但是 JDK 未提供 Java 源代码的编写环境,这个是 SUN 提供的很多基础开发工具的通病,所以实际的代码编写还需要在其他的文本编辑器中进行。其实大部分程序设计语言的源代码都是一个文本文件,只是存储的后缀名不同罢了。

常见的适合 Java 的文本编辑器有很多,例如 JCreator、Editplus、UltraEdit 等。下面依次来介绍 JDK 的下载、安装和环境配置。

1. JDK 的下载

如果需要获得最新版本的 JDK,可以到 SUN 公司的官方网站上进行下载,下载地址为:
<http://java.sun.com/javase/downloads/index.jsp>

下载最新版本的 JDK 6 Update 6,选择对应的操作系统,以及使用的语言即可。

2. JDK 的安装

Windows 操作系统上的 JDK 安装程序是一个可执行程序,直接安装即可,在安装过程中可以选择安装路径以及安装的组件等,如果没有特殊要求,选择默认设置。程序默认的安装路径在 C:\Program Files\Java 目录下。

3. JDK 环境变量的配置

Java 需要设置的环境变量有 3 个。

(1) 变量名 JAVA_HOME; 变量值 “C:\Program Files\Java\jdk1.4.2_04”。

变量值为你安装的 JDK 路径,在该路径下你应该能够找到 bin、lib 等目录,请根据自己的实际情况填写。本例中的设置就应为 “JAVA_HOME=C:\Program Files\Java\jdk1.4.2_04;”。

注意:值后面不要加分号。

(2) 变量名 PATH; 变量值 “%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;”。

设置 PATH 的目的是为了设置指向 JDK 的 bin 目录,在 bin 目录下存放的是各种编译执行命令。本例中的设置就应该为 “PATH=%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;”。需要说明的是,系统本身就有 PATH 环境变量,只要把 “%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;” 直接放到后面即可,中间使用 “;” 隔开。

(3) 变量名 CLASSPATH; 变量值 “.;%JAVA_HOME%\bin;%JAVA_HOME%\lib\tools.jar;”。

CLASSPATH 设置类的路径,十分重要,所有环境变量配置好后,在命令窗口中直接执行 java 或者 javac 命令,出现相应的信息,说明配置成功。配置环境变量步骤如下:

(1) 右击“我的电脑”图标,单击“属性”命令,选择“高级”选项卡,单击“环境变量”按钮,如图 1-1 所示。

(2) 在“系统变量”区域中,设置 3 项属性, JAVA_HOME, path, Classpath (大小写不影响)。若已存在则单击“编辑”按钮,不存在则单击“新建”按钮,如图 1-2 所示。

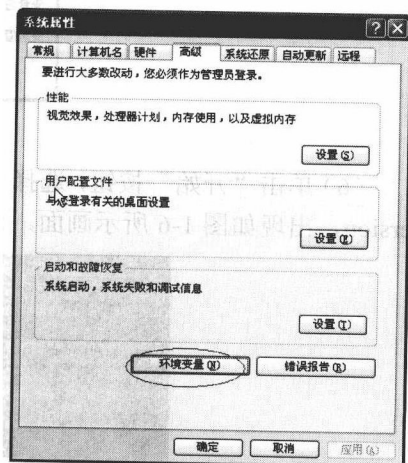


图 1-1 “系统属性”对话框

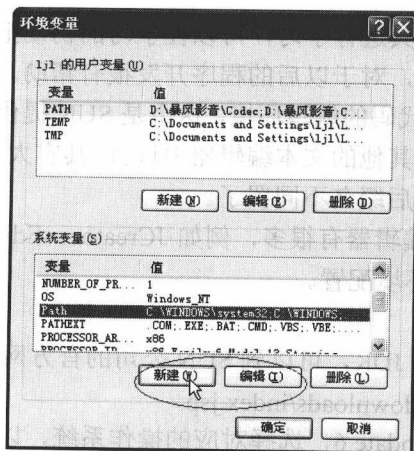


图 1-2 “环境变量”对话框

(3) JAVA_HOME 变量指明 JDK 安装路径, C:\Program Files\Java\jdk1.4.2_04, 就是刚才安装时所选择的路径, 此路径下包括 lib、bin、jre 等文件夹, 如图 1-3 所示。

(4) path 变量使系统可以在任何路径下识别 Java 命令, 值设置如图 1-4 所示。

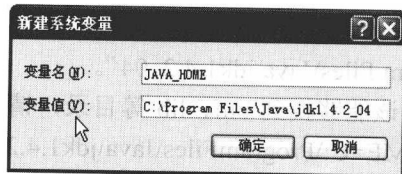


图 1-3 设置 JAVA_HOME

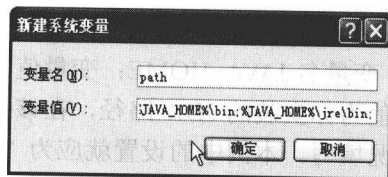


图 1-4 设置 path

(5) Classpath 为 Java 加载类 (class 或 lib) 路径, 只有类在 Classpath 中, Java 命令才能识别, 设为 “.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar” (要加 “.” 表示当前路径)。“%JAVA_HOME%” 就是引用前面指定的 JAVA_HOME, 如图 1-5 所示。

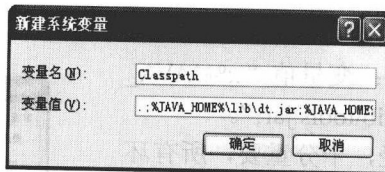


图 1-5 设置 Classpath

(6) 单击“开始”按钮, 选择“运行”项, 键入 cmd, 在命令提示符窗口中键入命令 java -version, 出现如图 1-6 所示画面, 说明环境变量配置成功。

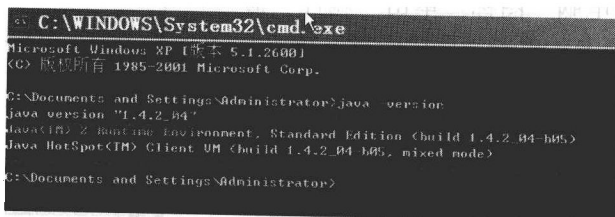


图 1-6 命令提示符窗口

1.1.6 Java Application 程序结构及运行过程

下面编写一个 Java 应用程序，通过实例来掌握 Java Application 的结构及运行过程。

1. 程序代码

【例 1-1】 HelloWorld

```
public class HelloWorld {                                // 声明一个公有类: HelloWorld
    public static void main(String[] args) {             // 类中主方法, 程序入口点
        System.out.println("Hello World !");            // 在屏幕上输出字符串 "Hello World !"
    }
}
```

运行结果如下:

Hello World !

2. 程序解析

(1) 每个 Java 程序至少包含一个类的声明。一个 Java 类由类的声明部分和类体两部分组成。

例 1-1 程序代码中的第 1 行 “public class HelloWorld {” 开始了一个公有类 HelloWorld 的声明。

- class 关键字引出 Java 的类声明，其后面直接跟上类名 HelloWorld。
- public 关键字用来修饰类，表示类的访问权限是公有的，对于公有的类则可以被其他任意类使用。
- 类体部分则由大括号 “{}” 括起来，此处只定义一个 main() 成员方法。

(2) 成员方法由方法声明和方法体两部分构成。

例 1-1 程序代码中的第 2 行 “public static void main(String argv[])” 为主方法 main() 的声明部分。

- public 关键字用来修饰方法，表示方法的访问权限是公有的，对于公有的方法则可以被其他任意类调用。
- static 关键字指明该方法是一个类方法，它可以通过类名直接调用，由 JVM 要求通过类直接调用 main() 方法，所以 main() 方法必须声明为类方法（static 方法）。
- void 则指明 main() 方法的返回值类型为空值（不返回值）。
- main() 方法首部圆括号 () 中的 String args[] 是传递给 main() 方法的参数，参数名为 args[]，但其名字允许改变，[] 亦可置于 args 与 String 之间，它是一个 String 类型的一维数组引用，它可以接受命令行中的 0 个或多个字符串类型的实际参数。
- main() 方法是运行 Java 应用程序的入口点，含有 main() 方法的类通常称为主类。

(3) 标准输出流。

main() 方法体部分由大括号 “{}” 括起来，此处 main() 方法体中只调用了 “System.out.println(“Hello World !”);” 这一条语句用来在显示器屏幕上输出一行字符串文本信息；System 是一个预定义的类，它提供对系统的访问。

- Out 是 System 类中的一个 PrintStream 类型的类成员变量，它代表标准输出流对象，将输出流传送到屏幕上。