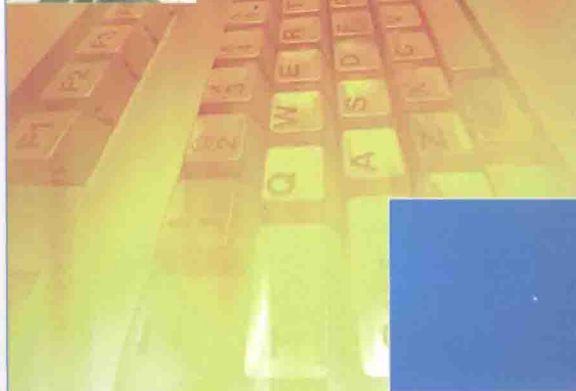


全国高职高专计算机技能型人才培养系列规划教材

中国计算机学会教育委员会
高职高专学组推荐教材

Java

程序设计基础



主 编 苏传芳
副主编 朱士明



北京大学出版社
PEKING UNIVERSITY PRESS

全国高职高专计算机技能型人才培养系列规划教材

(中国计算机学会教育委员会高职高专学组推荐教材)

Java 程序设计基础

主 编 苏传芳

副主编 朱士明



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书针对高职院校学生的特点,采用任务驱动的方式组织编写,实训案例丰富,易学好用。全书共11章,主要内容包括,Java语言概述,Java语言基础,数组,字符串,面向对象程序设计,异常处理,输入/输出处理,Java Applet,图形用户界面设计,多线程编程技术,最后一章“文档编辑器实例开发”是一个Java项目的综合实践案例,以提高学生Java编程的综合应用能力。

本书适合高职高专学生作为教材,也可作为本科及其他编程人员的Java入门课本或参考书。

图书在版编目(CIP)数据

Java 程序设计基础/苏传芳主编. —北京:北京大学出版社, 2008.9

(全国高职高专计算机技能型人才培养系列规划教材)

ISBN 978-7-301-14183-0

I. J… II. 苏… 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(2008)第 129359 号

书 名: Java 程序设计基础

著作责任者: 苏传芳 主编

策划编辑: 乐和琴

责任编辑: 孙琳

标准书号: ISBN 978-7-301-14183-0/TP·0969

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.com>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电子邮箱: pup_6@163.com

印 刷 者: 北京飞达印刷有限责任公司

发 行 者: 北京大学出版社

经 销 者: 新华书店

787mm×1092mm 16 开本 18.25 印张 460 千字

2008 年 9 月第 1 版 2008 年 9 月第 1 次印刷

定 价: 29.00 元

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有 侵权必究

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

《全国高职高专计算机技能型人才培养系列规划教材》 编委会委员名单

主任委员：俞光昀

副主任委员：刘乃琦

委 员 (按姓名拼音排序):

卜锡滨	陈书谦	程 刚	崔剑波
董汉丽	顾 滨	郝 梅	何文华
贺 平	蒋方纯	连卫民	梁锦叶
刘甫迎	刘湘涛	李金祥	骆耀祖
聂 明	宋汉珍	苏传芳	田绍愧
佟伟光	王 伟	王秀平	吴小惠
谢 尧	徐建民	严学道	杨丽芳
杨 威	杨学全	袁启昌	张凌雯
朱乃立			

秘 书 长：张荣琴 孙 琳

出版说明

高技能人才是国家核心竞争力的体现，加快高技能人才的培养已经纳入国家人才强国战略的总体部署。正是国家急需高技能人才的客观要求推动了高等职业教育的飞速发展。今天，论学生人数！高职高专学生已经占据了高等教育的半壁江山。每年几百万新生的招生规模是一个何等惊人的数字，将如此众多的青年人培养成具有良好的道德素养、熟练的职业技能的高技能人才是多么伟大的工程！对于肩负着这一伟大使命的高职高专院校，既是难得的历史机遇，又是艰辛的任务和挑战。我们要从不断改革教学模式、教学方法等各方面努力，争取把我国的高职高专教育推向一个新的高度。

在这样伟大的历史任务面前，中国计算机学会教育委员会高职高专学组和北京大学出版社联手成立了《全国高职高专计算机技能型人才培养系列规划教材》编委会，致力于从教材编写的角度为培养高技能人才做出新贡献。

二十多年前，由全国几十所大专、成人高校、电视大学、职工大学和夜大等大专层次的学校在湖南长沙发起成立了“全国大专计算机教育研究会”，1986年全国大专计算机教育研究会加入中国计算机学会教育委员会，简单称大专学组，从此就在中国计算机学会教育委员会的指导下有计划地开始了大专层次的计算机专业的教育和教材建设的研究。同年，经原电子工业部批准，在全国大专计算机教育研究会的基础上，成立了“全国大专计算机专业教材编委会”。随着高职高专教育的发展，随着新世纪的来临，大专学组和全国大专计算机专业教材编委会分别更名为高职高专学组和全国高职高专计算机专业教材编委会。

二十多年来，高职高专学组和高职高专计算机专业教材编委会一方面不断研究改进高职高专计算机各专业方向的培养计划和教学方法，另一方面与出版社合作联合成立相关系列编委会致力于高职高专计算机专业系列教材的编写工作。二十多年来，共完成了五轮近三百种教材的编写工作。

计算机高职高专教材的出版，解决了大专计算机教学依赖本科教材的局面，一轮又一轮，一批又一批教材的相继出版，不但使高职高专教材的质量与时俱进，同时还推动了高职高专院校师资队伍的成长。

但是，由于我国职业教育起步较晚，至今还没有形成西方发达国家那样完整的职业教育体系，因此在职业教育的许多方面，包括教材建设方面还存在着相对落后的方面和诸多不足。就教材而言，存在着部分新专业没有教材；教材内容陈旧，不适应新技术发展的需要；实践技能教材严重缺乏；教材内容和职业资格证书制度衔接不足等。

我国社会主义现代化建设需要大批高技能人才，而高技能人才的培养需要科学的、合理的教材。《全国高职高专计算机技能型人才培养系列规划教材》旨在教材建设中引进国内外成熟的经验，同时适应高等职业教育不断改革的需要，在教材内容和教材风格上有所创新。

本套教材计划按照每门课程的不同特点，分别采用任务驱动法、项目教学法或案例教学法。

在教材内容上，本套教材力图将最新的知识、最新的技术写进教材；着重讲解技能型人才培养所需的内容和关键点，突出实用性和可操作性；尽量采用综合性的实例来讲解理论知识。

识的综合运用，“以例释理”，将理论讲解简单化，从而锻炼学生的思维能力以及运用概念解决问题的能力；要设计具备真实性的实践操作训练项目，加强学生对工程实践的兴趣，提高他们的实践操作技能；为了满足学有余力的学生深入学习的需要，我们提倡模块化编写方法，有些科目需要编写提高模块。

在编写风格上，本套教材将努力学习和借鉴国内外优秀教材的写作思路、写作方法和章节安排；作为工科教材，本套教材也将借鉴人文学科教材的写作模式，体现清新活泼的风格；部分教材还将采用学校教师任主编，企业高工任主审的方式，依托行业和企业共同进行编写；在出版纸质教材的同时，还将编写网络课件、CAI 课件、教学素材库、电子教案、试题库及考试系统和多媒体教学软件。

本套教材不仅适合高职高专院校计算机及相近专业的学生使用，也适用于企事业单位从业人员的在职培训，对于社会上广大自学人员的素质提高也具有实用价值和参考作用。

中国计算机学会教育委员会高职高专学组
《全国高职高专计算机技能型人才培养系列规划教材》编委会
2008年8月

前 言

开拓一种适合高职教育教学模式的教材是编写本书的宗旨。本书总结了编者多年的高职教学经验,采用“傻瓜式教学”的思想:先动手后动脑,趣味性教学。由浅入深的层次结构组织教材内容,做到以实例为先导,内容新颖,注重在应用中学习,起到了引人入胜,事半功倍的效果。

本书针对高职教育的特点,具有以下鲜明的特色。

着重体现任务驱动的教学特点。Java 语言中面向对象的编程思想是关键所在,对于高职学生,学习起来难度较大。我们以精选案例为主导,在案例教学中贯穿理论教学,先动手后动脑,以应用为目标学习理论知识,重在提高学生的技术应用能力;强调实践环节,每章节都配有实训或课题设计,有明确的学习任务。

实训案例丰富:从知识点的理解、知识技能的掌握到综合知识的应用,由浅入深,以提高学生的总体素质。使用本教材的老师,也会大大减小备课的工作量,不必费心去寻找实例程序。

实训内容有:思路分析、解题方法,要点总结。学生可以按照所给的程序去做,上手快,立竿见影。既培养了学生学习兴趣,又消除了对设计程序的畏惧心理,每个人都可以顺利完成实训。

最后一章的内容为一个具体的应用开发实例,通过本章的训练,进一步巩固、提高所学的知识 and 技能,确保学生“学有所用,学而能用”。

教学效果表明,学生在实训中自己动手,在干中学习,非常投入,在实践中不知不觉地提高能力,走上良性学习的轨道。

本书较适宜的授课学时为 90 学时左右,各章的参考教学时数如下表。

章 次	建议学时	章 次	建议学时
第 1 章 Java 语言概述	4	第 7 章 输入/输出处理	8
第 2 章 Java 语言基础	10	第 8 章 Java Applet	6
第 3 章 数组	4	第 9 章 图形用户界面设计	16
第 4 章 字符串	4	第 10 章 多线程编程技术	8
第 5 章 面向对象程序设计	16	第 11 章 Java 项目实践——文档编辑器实例开发	8
第 6 章 异常处理	6		

总学时 90(理论教学:实训教学=1:1)

本书由安徽电子信息职业技术学院苏传芳任主编,朱士明任副主编。全书由安徽电子信息职业技术学院教师编写,其中第 1、6、8 章由朱士明编写;第 2 章由苏传芳编写;第 5、11 章由朱正月编写;第 3、4、7 章由张红梅编写;第 9、10 章由施艳昭编写。苏传芳负责全书的总体设计和统稿。

编 者
2008 年 5 月

目 录

第 1 章 Java 语言概述	1
1.1 Java 的发展与应用.....	1
1.2 Java 的开发工具与开发环境.....	2
1.2.1 JDK 的下载与安装.....	2
1.2.2 JDK 的环境配置.....	3
1.2.3 JDK 的主要开发工具.....	3
1.3 Java 程序的初步了解.....	4
1.3.1 编写 Java 源程序.....	5
1.3.2 编译 Java 源程序.....	6
1.3.3 运行 Java 程序.....	7
1.4 Java 的工作机制和基本特点.....	8
1.4.1 Java 的工作机制.....	8
1.4.2 Java 的基本特点.....	9
实训 1 Java 程序的运行.....	10
本章小结.....	11
习题.....	11
第 2 章 Java 语言基础	12
2.1 Java 程序的基本构成要素.....	12
2.1.1 标识符.....	12
2.1.2 关键字.....	13
2.1.3 分隔符.....	13
2.2 数据类型.....	14
2.2.1 基本数据类型.....	14
2.2.2 常量与变量.....	15
2.3 运算符与表达式.....	17
2.3.1 算术运算符与算术表达式.....	17
2.3.2 关系运算符与关系表达式.....	19
2.3.3 逻辑运算符与逻辑表达式.....	19
2.3.4 位运算.....	21
2.3.5 赋值运算符和赋值表达式.....	22
2.3.6 三元条件运算符.....	23
2.3.7 其他运算符.....	24
2.4 表达式.....	24
2.4.1 常用数值计算方法.....	24
2.4.2 类型转换.....	25
2.4.3 优先级和结合性.....	27
2.5 包装类.....	28
2.6 输入/输出处理.....	30
2.6.1 输出流.....	30
2.6.2 输入流.....	30
实训 2 Java 的输入/输出编程.....	31
2.7 方法的使用.....	32
2.7.1 方法的定义与调用.....	32
2.7.2 方法调用中数据的传递.....	33
2.7.3 方法和变量的作用域.....	34
2.7.4 方法的嵌套和递归调用.....	35
2.7.5 方法的重载.....	36
实训 3 嵌套和递归调用.....	37
2.8 流程控制.....	37
2.8.1 语句.....	38
2.8.2 分支语句.....	39
实训 4 if 语句和 switch 语句的使用.....	45
2.8.3 循环语句.....	47
2.8.4 标号和其他流程控制语句.....	53
实训 5 while, do-while 和 for 语句 的使用.....	54
课程设计 1 设计一个简单计算器.....	55
本章小结.....	55
习题.....	56
第 3 章 数组	60
3.1 一维数组.....	60
3.1.1 一维数组的声明.....	60
3.1.2 一维数组的空间分配.....	61
3.1.3 一维数组的使用.....	62
3.1.4 数组作为方法的参数和 返回值.....	64
实训 6 while, do-while 和 for 语句的 使用.....	65

3.2 二维数组.....	67	5.4.2 接口的实现.....	121
3.2.1 二维数组的声明.....	67	5.4.3 接口的应用.....	122
3.2.2 二维数组的空间分配.....	67	5.5 包(Package).....	125
3.2.3 二维数组的使用.....	68	5.5.1 包的概述.....	125
实训 7 二维数组的应用.....	69	5.5.2 包的创建.....	126
本章小结.....	71	5.5.3 包的引入.....	128
习题.....	72	5.5.4 类与类成员的访问保护.....	131
第 4 章 字符串	74	实训 12 包的创建、使用与访问保护.....	132
4.1 String 字符串类.....	74	课程设计 2 定义教师类, 计算教师	
4.1.1 类 String 字符串的创建.....	74	工资.....	135
4.1.2 类 String 字符串处理的常用		本章小结.....	138
方法.....	76	习题.....	139
实训 8 String 类的应用.....	80	第 6 章 异常处理	141
4.2 StringBuffer 字符串类.....	81	6.1 异常的概述.....	141
4.2.1 类 StringBuffer 字符串的		6.2 Java 中的异常类.....	142
创建.....	81	6.2.1 异常类的层次结构.....	142
4.2.2 类 StringBuffer 字符串的常用		6.2.2 Throwable 类.....	143
方法.....	82	6.3 异常处理机制.....	144
4.3 命令行参数.....	85	6.4 异常的捕获与处理.....	144
实训 9 类 String 和 StringBuffer 的		6.4.1 使用 try-catch 语句.....	145
比较.....	86	6.4.2 多个 catch 子句结构.....	145
本章小结.....	88	6.4.3 finally 子句.....	146
习题.....	88	6.5 throw 和 throws 语句.....	147
第 5 章 面向对象程序设计	90	6.5.1 throw 语句.....	147
5.1 面向对象程序设计概述.....	90	6.5.2 throws 语句.....	148
5.2 类(Class).....	91	6.6 定义自己的异常类.....	149
5.2.1 类的声明.....	91	实训 13 异常处理的应用.....	151
5.2.2 类的对象.....	93	本章小结.....	153
5.2.3 特殊的方法——构造方法.....	95	习题.....	153
5.2.4 类和成员的修饰符.....	99	第 7 章 输入/输出处理	156
5.2.5 类的使用.....	101	7.1 输入/输出流概述.....	156
实训 10 类的定义与应用.....	107	7.2 输入/输出类.....	157
5.3 类的特性.....	109	7.2.1 字节流 InputStream 类和	
5.3.1 继承性.....	109	OutputStream 类.....	158
5.3.2 多态性.....	115	7.2.2 字符流 Reader 类和	
实训 11 继承与多态的实现及应用.....	118	Writer 类.....	159
5.4 接口(Interface).....	119	7.2.3 标准输入/输出.....	160
5.4.1 接口的定义.....	120	7.3 文件的顺序访问.....	161
		7.3.1 输入/输出流操作的一般	
		步骤.....	161

7.3.2 字节流类.....	162	9.2.2 容器类组件.....	199
7.3.3 字符流(Reader 类和 Writer 类).....	167	9.2.3 控制组件.....	201
实训 14 文件输入/输出.....	170	9.3 指定布局管理器.....	202
课程设计 3 一个简单的学生信息录入 程序.....	171	9.3.1 FlowLayout.....	202
7.4 文件的随机访问.....	172	9.3.2 BorderLayout.....	203
7.4.1 建立随机访问文件流对象.....	172	9.3.3 GridLayout.....	205
7.4.2 读/写随机访问文件方法.....	173	9.3.4 CardLayout.....	206
7.4.3 文件指针及相关方法.....	173	9.3.5 GridBagLayout.....	207
7.5 目录和文件管理.....	174	9.4 事件处理机制.....	209
7.5.1 目录管理.....	174	9.4.1 带有事件处理的 GUI 程序....	209
7.5.2 文件管理.....	174	9.4.2 事件处理.....	210
实训 15 文件.....	176	9.4.3 文本事件.....	216
本章小结.....	177	9.4.4 鼠标事件.....	217
习题.....	177	9.4.5 键盘事件.....	218
第 8 章 Java Applet	179	9.5 常用的组件.....	219
8.1 Java Applet 的特点.....	179	9.5.1 按钮及标签.....	219
8.2 Java 与 HTML 的关系.....	180	9.5.2 Swing 中的框架.....	219
8.2.1 HTML 语言.....	180	9.5.3 JApplet 类.....	221
8.2.2 在 HTML 中嵌入 Applet.....	182	9.5.4 画布.....	222
8.2.3 Applet 的执行.....	183	9.5.5 JTabbedPane 类.....	223
8.3 Applet 的创建与生命周期.....	183	9.5.6 JTree 组件.....	225
8.3.1 Applet 类的层次及框架 结构.....	183	9.5.7 JCheckBox 类.....	226
8.3.2 Applet 的主要方法与生命 周期.....	184	9.5.8 单行文本框.....	227
8.4 HTML 与 Applet 之间的参数传递....	186	9.5.9 颜色选择器(JColorChooser) ..	228
实训 16 Applet 的编写与运行.....	188	9.6 创建和使用菜单.....	229
本章小结.....	190	9.6.1 创建菜单条式菜单.....	230
习题.....	190	9.6.2 菜单子项定义快捷键.....	232
第 9 章 图形用户界面设计	192	9.6.3 创建菜单.....	232
9.1 GUI 概述.....	192	9.7 观感.....	236
9.1.1 AWT 简介.....	193	实训 17 通信簿.....	236
9.1.2 Swing 简介.....	195	本章小结.....	238
9.1.3 组件.....	196	习题.....	238
9.2 组件的创建与使用.....	198	第 10 章 多线程编程技术	239
9.2.1 Component 类.....	198	10.1 概述.....	239
		10.1.1 进程与线程.....	240
		10.1.2 线程的概念模型.....	240
		10.1.3 实现线程的类(Thread 类)....	241
		10.1.4 线程的状态.....	242

10.2 创建和启动线程.....	243	11.1.3 主控窗口设计.....	251
10.2.1 创建线程.....	244	11.1.4 弹出式菜单.....	261
10.2.2 启动线程.....	245	11.2 事件处理.....	263
实训 18 动态数字时钟.....	248	11.2.1 主菜单项事件处理.....	263
本章小结.....	249	11.2.2 工具栏按钮事件处理.....	271
习题.....	249	11.2.3 弹出式菜单项事件处理.....	273
第 11 章 Java 项目实践——文档编辑器		11.3 核心类处理代码.....	273
实例开发	250	本章小结.....	276
11.1 文档编辑器功能结构设计.....	250	参考文献	277
11.1.1 功能结构.....	250		
11.1.2 主窗口外观与操作设计.....	251		

第 1 章 Java 语言概述

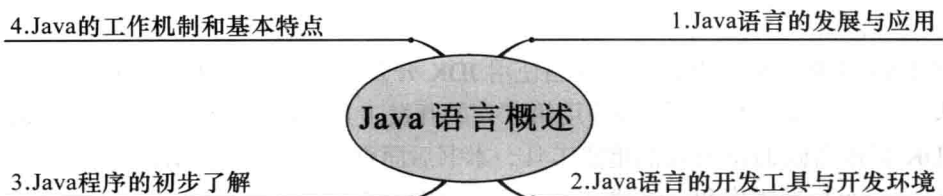


本章要点

Java 是面向对象的编程语言，也是当前最流行的网络编程语言。Java 的出现引起了软件开发的变革，为迅速发展的 IT 业增添了新的活力。本章的主要任务是了解 Java 的发展、应用与特点，掌握 Java 开发工具 Java JDK 软件包的下载和开发环境 JDK 的设置，了解 Java 程序的概貌及其运行方法。



本章概要



1.1 Java 的发展与应用

Java 是 Sun 公司于 20 世纪 90 年代初开发的。最初并不是为了用于 Internet，而是作为一种小家用电器的编程语言，用来解决诸如电视机、电话、闹钟、烤面包机等家用电器的控制和通信问题，命名为 Oak。由于这些智能化家用电器的市场需求当时没有预期的高，Sun 放弃了该项计划。就在 Oak 几近夭折之时，Internet 异常火爆起来。Sun 看到了 Oak 在计算机网络上的广阔应用前景，他们改造了 Oak，将 Oak 技术应用于 Web 上，并且开发出了 HotJava 的第一个版本，并于 1995 年 5 月发表，在产业界引起了巨大的轰动，Java 的地位也随之得到肯定。又经过一年的试用和改进，Java1.0(JDK1.0)终于在 1996 年初正式发布，从而引发了遍及全球、至今未衰的 Java 热潮。用 Java 可以编写嵌入在 Web 网页中运行的 Java Applet 小程序，也可以编写独立运行的 Java Application。Java 是当今十分流行的网络编程语言。Java 语言的出现被称作是一场软件革命。

Java 伴随着 Internet 的迅猛发展而发展，逐渐成为重要的 Internet 编程语言。由于 Java 提供了强大的图形、图像、动画、音频、多线程及网络交互能力，使它在设计交互式、多媒体网页和网络应用程序方面大显身手，成为当今推广速度最快的一门计算机程序语言。

Java 语言正在不断发展和完善，Sun 公司是主要的发展推动者。近两年 Sun 公司推出的三大平台 J2SE、J2ME、J2EE 显现了 Java 语言强大的应用能力，从高性能计算到移动计算(手机通信)、智能卡，到处都有 Java 的身影。

Java 具有“一次编程，到处运行”的独特之处，用 Java 语言开发的软件可以不用修改或重新编译而直接应用于任何计算机上，极大地提高了软件开发的效率。从而使其具有广泛的应用前景，主要有以下几个方面的应用：

- (1) 计算过程的可视化、可操作化的软件开发；
- (2) 动态画面的设计，包括图形图像的调用；
- (3) 所有面向对象的应用开发，包括面向对象的事件描述、处理、综合等；
- (4) 交互操作的设计(选择交互、定向交互、控制流程等)；
- (5) Internet 的系统管理功能模块的设计，包括 Web 页面的动态设计、管理和交互操作设计等；
- (6) Intranet(企业内部网)上的软件开发(直接面向企业内部用户的软件)；
- (7) 与各数据库连接查询的 SQL 语句实现；
- (8) 其他应用类型的程序。

1.2 Java 的开发工具与开发环境

许多 Java 专家认为，初学 Java 应当使用 JDK 开发环境。虽然 JDK 的界面不如一些可视化工具友好，许多编程人员通常都使用第三方的开发工具，但它却是各种开发环境的基础。因此，JDK 仍被当做 Java 开发的重要工具。本书后面章节将以介绍 JDK 为主。

JDK 由一个标准类库和一组建立、测试及建立文档的 Java 实用程序组成，其核心 Java API(应用程序接口)是一些预定义的类库，开发人员需要用这些类来实现 Java 语言的功能。Java API 包括一些重要的语言结构以及基本图形、网络和文件的输入/输出(简记为 I/O)。一般来说，Java API 的非 I/O 部分对于运行 Java 的所有平台是相同的，而 I/O 部分则仅在通用 Java 环境中实现。

1.2.1 JDK 的下载与安装

为了建立基于 JDK 的 Java 开发环境，需要先下载 Sun 公司的免费 JDK 软件包。JDK 包含了一整套开发工具，其中包含对编程最有用的 Java 编译器、Applet 查看器和 Java 解释器等工具。

从 Sun 公司的 Internet 站点可以找到最新版本的 JDK，也可以从很多镜像站点下载 JDK，但需要注意这些镜像站点所保存的 JDK 的版本。下载网址，如 <http://java.sun.com/> 为 Sun 公司网站页面。

在页面右边 Popular Downloads 栏中可看到最新版的 JDK 软件包，这里显示为“J2SE1.6.0 JDK”(当前版本)。

打开“J2SE1.6.0JDK”超链接，单击“Windows Offline Installation, Multi-language”超链接，开始下载 `jdk-6-windows-i586(1).exe`。

运行下载的“`jdk-6-windows-i586(1).exe`”软件包，可安装 JDK，在安装过程中可以设置安装路径及组件，系统默认的安装路径为：`C:\Program Files\Java`，默认组件选择是全部安装。

1.2.2 JDK 的环境配置

在使用 JDK 以前，必须配置环境变量。下面以 Windows 2000/XP 系统为例说明。

用鼠标右键单击“我的电脑”，选择“属性”，在系统属性对话框中，单击“高级”按钮，单击“环境变量”按钮，打开“环境变量”对话框，如图 1.1 所示，找到变量名“Path”，选中后，单击“编辑”按钮，增加路径“C:\Program Files\Java\jdk1.6.0\bin”，如图 1.2 所示；新建变量名“CLASSPATH”，其值为“.”；”，如图 1.3 所示。



图 1.1 找到变量名(Path)

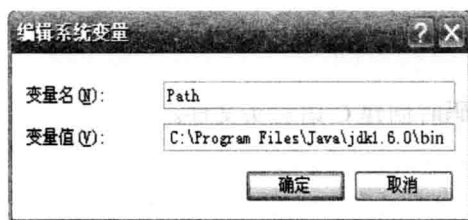


图 1.2 增加路径

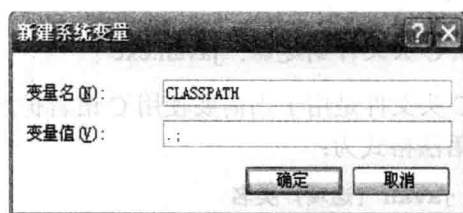


图 1.3 新建变量名

保存后，打开一个新的 DOS 窗口，输入 java 命令，发现已经生效了。

1.2.3 JDK 的主要开发工具

Java JDK 包括了所有编译、运行 Java 程序所需要的工具：Java 基本组件、库、Java 编译器、Java 解释器、小应用程序浏览器等。这些工具都在 Java JDK 安装目录下的 bin 子目录里，以下为一些基本的工具。

1. Java 编译器 javac.exe

Java 编译器将以“.java”为扩展名的 Java 源程序文件编译成类文件(扩展名：.class)。

语法格式为：

```
javac [选项] 源程序文件名  
源程序文件名的扩展名.java 不能省略。
```

例如：javac Hello.java

2. Java 解释器 java.exe

Java 解释器用于执行编译过的 Java 应用程序的类文件。

语法格式为：

```
java [选项] 类名  
类文件名的扩展名.class 可略去。
```

例如：java Hello

3. Applet 浏览器:appletviewer.exe

Applet 浏览器用于运行 Java Applet 程序。

语法格式为：

```
appletviewer [选项] URL
```

URL 表示 URL 描述的 HTML 文档，要指出文件的扩展名 html 等。

例如：appletviewer HelloApplet.html

4. Java 反编译器: javap.exe

Java 反编译器用于显示编译后的字节码。

语法格式为：

```
javap [选项] 类名
```

5. C 头文件创建器: javah.exe

C 头文件是用于当需要使用 C 语言扩展 Java 代码时创建 C 语言头文件。

语法格式为：

```
javah [选项] 类名
```

6. Java 文档生成工具: javadoc.exe

Java 文档生成工具是用于在 Java 源文件中生成基于标志的 HTML 文件。

7. Java 调试器: jdb.exe

Java 调试器是 Java 环境的调试工具。

1.3 Java 程序的初步了解

Java 程序的开发环境和运行环境准备好以后，就可以开始编写 Java 程序。要编写 Java 程序，需要有文本编辑器。可以使用操作系统提供的 Edit 或记事本作为编辑器，也可选择 EditPlus

编辑软件，它使用不同颜色显示 Java 的关键字和类名，简单好用。

1.3.1 编写 Java 源程序

Java 程序分为应用程序(Application)和小程序(Applet)，下面分别进行介绍。

1. Java 应用程序举例

【例 1.1】 在屏幕上显示文字“欢迎你学习 Java 语言！”的应用程序。

打开一个文本编辑器，输入以下代码：

```
/*Hello.java*/
public class Hello{           //一个 Application
    public static void main(String args[ ]){
        System.out.println(" 欢迎你学习 Java 语言!");
    }
}
```

在源程序中的第 1 行用“/*”和“*/”括起来以及第 2 行后面以双斜线“//”引导的内容是 Java 语言的注释信息。在程序中使用注释，可增加程序的可读性。

第 2 行开始是类的定义，设计任何 Java 程序必须声明类。其中保留字 class 用来定义一个新的类，其类名是 Hello，它是一个公共类(public)。Java 程序中可以定义多个类，但是最多只有一个公共类，并且程序文件名要求与该公共类的类名相同。整个类定义由大括号{ }括起来，其内部称为类体，类体中定义类的数据成员和成员方法。

第 3 行定义了类的方法 main()，其中 public 表示访问权限，指明所有的类都可以使用这一方法；static 指明该方法是一个类方法，它可以通过类名直接调用；void 则指明 main()方法不返回任何值。对于 Java 应用程序而言，main()方法是必需的，以它作为应用程序执行的入口，而且必须按照上述格式来定义。

第 4 行是 main()方法体的具体内容，其功能是输出字符串。

当输入代码完成后，根据环境变量设置的路径，将文件保存为文本格式，文件名要取 Hello.java。

2. Java 小程序举例

【例 1.2】 显示文字“欢迎你学习 Java 语言！”的小程序。

```
/*HelloApplet.java*/
import java.awt.Graphics;
import java.applet.Applet;
public class HelloApplet extends Applet{ //一个 Applet
    public void paint(Graphics g){
        g.drawString("欢迎你学习 Java 语言!", 20, 20);
    }
}
```

在这个小程序中，首先用 import 语句引入 java.awt.Graphics 类和 java.applet.Applet 类，这是本小程序需要的两个类：小程序需要继承 Applet 类来构造；图形界面的小程序输出常用具有绘图功能的 paint()方法，该方法需要 Graphics 类的参数，然后定义一个公共类 HelloApplet，用 extends 指明它是 Applet 的子类。Java 小程序都是 Applet 和 JApplet 类的子类。在类体中重写父类 Applet 的 paint()方法，其中参数 g 为 Graphics 类的对象，可认为是用于绘图的画板。

在 `paint()` 方法中, 调用对象 `g` 的方法 `drawString()`, 在坐标(20,20)处输出字符串“欢迎你学习 Java 语言!”, 其中坐标是以像素点为单位的。

当输入代码完成后, 根据环境变量设置的路径, 将文件保存为文本格式, 文件名要取 `HelloApplet.java`。

注意: Application 程序中含有 `main()` 方法, 是程序执行的入口, 一般在本地或服务器上由 Java 解释器解释并执行; 而 Applet 程序中没有 `main()` 方法, 由小程序查看器或含有 Java 解释器的浏览器下载并执行, 这是 Applet 与 Application 主要区别之一。

3. Java 程序结构

从上述例子可以看出, Java 程序是由类构成的, 对于应用程序来说, 必须在一个类中定义有 `main()` 方法, 包含 `main()` 方法的类是该应用程序的主类; 对于小程序来说, 它必须作为 Applet 类或 JApplet 类的一个子类, 继承 Applet 类或 JApplet 类的类是小程序的主类。

Java 源程序的书写规则需要注意以下几点。

(1) 文件命名规则。若 Java 文件包含一个 `public` 类, 则一定要以该类的名称来命名文件。例如, 例 1.1 中 `public` 类型的类是 `Hello`, 源程序文件名必须命名为 `Hello.java`; 例 1.2 中 `public` 类型的类是 `HelloApplet`, 源程序文件名必须命名为 `HelloApplet.java`。且 Java 对字母的大小写是敏感的, 类名与程序文件名的大小写也要一致。源程序中定义的类名其首字母最好大写, 以区别变量。

(2) 类个数的规定。一个源文件中最多只有一个 `public` 类, 其他类的个数不限。

(3) 其他内容。

`package` 语句: 包语句, 0 个或 1 个, 必须放在文件开始位置。

`import` 语句: 导入语句, 0 个或多个, 必须放在所有类定义之前。

`interfaceDefinition`: 接口定义, 0 个或多个。

`public classDefinition`: `public` 类定义, 0 个或 1 个, 必须与文件同名。

`classDefinition`: 类定义, 0 个或多个。

在书写源程序时, 最好采用分层次缩进方式书写, 有助于增加程序的可读性。

1.3.2 编译 Java 源程序

Java 源程序编写后, 要使用 Java 编译器(`javac.exe`)进行编译, 将 Java 源程序编译成可执行的程序代码。

Java 源程序都是扩展名为 `java` 的文本文件。编译时首先读入 Java 源程序, 然后进行语法检查, 如果出现问题就中止编译, 显示错误信息。语法检查通过后, 生成可执行程序代码(字节码), 字节码文件名和源文件名相同, 但扩展名为 `.class`。

在命令提示符窗口或 MS-DOS 窗口进入源程序所在目录, 然后输入编译器文件名和要编译的源程序文件名。例如, 要编译例 1.1 中的源程序文件 `Hello.java`, 输入命令: `javac Hello.java`, 在命令执行过程中, 如果源程序没有错误, 则屏幕上没有输出, 这时会在相应的目录中生成一个同名字的 `class` 文件 `Hello.class`; 同样, 例 1.2 中的源程序文件 `HelloApplet.java` 经编译后会在目录中生成同名字的字节码文件 `HelloApplet.class`, 如图 1.4 所示。