

国家示范性高职院校建设规划教材

实用Java语言 单元式模块化教程

任泰明 郭领艳 主编 曾强 主审



化学工业出版社

国家示范性高职院校建设规划教材

实用 Java 语言单元式模块化教程

任泰明 郭领艳 主 编
曾 强 主 审

图例在版编目(CIP)数据

实用 Java 语言单元式模块化教程 / 任泰明, 郭领艳主编

北京: 化学工业出版社, 2008.6

国家示范性高职院校建设规划教材

ISBN 978-7-122-03312-4

I. 实... II. 任... III. Java 语言—教材

IV. TP312 计算机—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 102160 号

文字编辑: 李 林

责任编辑: 郭领艳

封面设计: 李 林

责任校对: 李 林

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 北京市兴源印刷厂

787mm×1092mm 1/16 印张 14.4 字 362 2008 年 9 月北京第 1 次印刷

购书咨询: 010-64218888 (传真: 010-64213688) 售后服务: 010-64218806



化学工业出版社

凡购本书, 如内缺页或质量问题, 本社负责调换。北京

本书以职业技术教育的特点出发,在内容组织与编写上更加突出实践技能与软件开发中工程化思想的培养。具体内容按 Java 语言软件开发过程中所需要的基本技能进行组织,全书共分为 10 个技术单元,每个单元分为若干个技术点,每个技术点以一个模块进行组织,每一个模块按案例教学法的特点与要求进行知识与技能介绍。全书主要讲述了 Java 语言程序设计的基本知识、类的设计技术、复杂数据类型的使用技术、软件的重用技术与接口的设计技术、程序中异常情况的处理技术、图形用户界面程序的设计技术、文件访问技术、用 Java 程序访问数据库的技术、多线程程序的设计技术等内容。

本书内容通俗易懂,以实例引出知识与技术点,非常适合自学和高职高专院校作为“Java 语言程序设计”课程的教材使用,也可供从事 Java 语言程序设计者作为参考书或 Java 语言程序设计培训机构作为培训教程使用。

实用 Java 语言单元式模块化教程/任泰明,郭领艳主编

主 编 任泰明 郭领艳

审 主 题 曾

图书在版编目(CIP)数据

实用 Java 语言单元式模块化教程/任泰明,郭领艳主编.

北京:化学工业出版社,2008.6

国家示范性高职院校建设规划教材

ISBN 978-7-122-03315-4

I. 实… II. ①任…②郭… III. Java 语言-程序设计-高等学校:技术学院-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 102160 号

责任编辑:张建茹

文字编辑:孙 科

责任校对:李 林

装帧设计:关 飞

出版发行:化学工业出版社(北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装:北京市兴顺印刷厂

787mm×1092mm 1/16 印张 14¼ 字数 362 千字 2008 年 9 月北京第 1 版第 1 次印刷

购书咨询:010-64518888(传真:010-64519686) 售后服务:010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书,如有缺损质量问题,本社销售中心负责调换。

定 价:25.00 元

版权所有 违者必究

前言

Java 语言自 1995 年产生以来,应用和普及的速度非常快,其应用领域已经从当初的桌面应用软件发展到目前大型企业管理系统的开发中,是人们当前公认的自程序设计语言产生以来发展速度最快、应用最广的高级程序语言。近年来的各种统计也表明,Java 语言已经成为程序员最喜欢的语言之一。在国内的高校中,已经有一些学校将 Java 语言作为学生的第一门程序设计语言课。这些都表明现在和今后相当长的一段时间内,Java 语言将是程序设计的主流语言之一,相应的与 Java 有关的技术成了当前程序设计领域最为热门的技术。Java 的广泛应用也使社会对掌握了 Java 技术的人才有较大的需求量,为此各类高职院校的计算机专业和 IT 培训机构也在近年来调整教学计划,把 Java 作为教学的重点内容。

由于 Java 语言的广泛应用,各出版社出版了很多 Java 语言教材,有些教材在内容的组织和编写上还是非常不错的,但就目前高职高专类 Java 语言教材而言,不同程度的存在着一些问题,这些问题也是本教材想努力改进的地方,从作者的教学实践来看主要有以下几点。

(1) 教材内容与教育部对高等职业技术教育的要求有一定的差距。教育部要求高等职业教育要“加大课程建设与改革的力度,增强学生的职业能力”。现出版的大量 Java 语言教材中,以知识点和语言体系组织教材内容,这种教材的优点是学生可以比较系统地学习程序设计语言的有关理论知识,但淡化了对学生职业能力的培养。

(2) Java 语言教材中程序的书写格式与编程风格不符合目前软件开发公司的一般编程规范。就 Java 语言来说,国内和国外的软件开发公司有一个行业领域比较通用的编程风格,如对程序代码的书写格式、标识符的命名习惯、注释的书写要求等。但目前在高职教材中将这些内容纳入其中的难得一见。

(3) 教育部对高等职业技术教育中实践技能较强的课,要求理论与实践学时基本要达到 1:1 的水平,教师在讲完有关理论和技能知识以后,学生在实践环节中无所适从(当然有些教材有配套的上机实践指导书),更重要的是人为地将理论与实践教学分成了两个部分。

(4) Java 语言是一种面向对象的程序设计语言,应该用面向对象的编程思想组织教材内容与编写教材中的实例程序。

本教材将学生通过 Java 语言的学习所要掌握的几个主要职业能力分为 10 个单元,每个单元根据学生所要掌握的具体职业技能分为若干个模块,每个模块以一个比较具有实践意义的小案例程序为例,引出本模块教学中所要掌握的知识与技能,

并展开对相关技能和知识的介绍与说明,在每个模块后有与本模块紧密相关的同步技能练习,技能练习的重点是实践环节的练习。学生在自学或教师在组织教学时,一个模块可以按照1到2个学时的时间进行组织与讲解,符合教学要求与自主学习特点,使学生在愉快中学习。本教程中的所有案例程序书写时,严格按有关行业规范进行。每一模块后的【技能同步练习】与该模块中的相关知识与技能密切相关,使学生在学完该模块后,通过编程实践进一步巩固所学的知识。

本书介绍的内容是Java技术中基本的J2SE技术,以J2SE 5.0为基础,所以程序均在J2SE5.0环境下调试。本书内容共分为10个单元。

本书在编写过程中充分考虑当前职业院校学生的知识水平和学习能力,尽量做到在语言文字上通俗、明了,案例难度适中,以便于教学和自学。

本书由任泰明、郭领艳主编。曾强主审。全书由任泰明组织编写并统稿,参与本书编写工作的有任泰明、郭领艳、穆德恒、韩永印、秦鹏翔、陆启军、李娟、田春婷、王海婴。另外,本书在编写过程中得到了化学工业出版社编辑的大力支持和帮助,也得到了宋贤钧教授的帮助,在此一并表示感谢。

由于作者水平有限,不足之处在所难免,敬请广大读者批评指正。

编者

2008年5月

目 录

第1单元 程序运行环境的建立与程序的编译调试技术	1
模块 1.1 Java 语言的产生	1
1.1.1 Java 语言的产生	1
1.1.2 用 Java 语言编写程序的主要优点	2
模块 1.2 建立 Java 编程环境	4
1.2.1 安装和设置 JDK	4
1.2.2 【技能同步练习】	8
模块 1.3 Java 程序的调试、编译与运行技术	8
1.3.1 【案例 1-1】HelloWorld 程序	8
1.3.2 【知识与技能】	11
1.3.3 【技能同步练习】	13
模块 1.4 Java 语言中注释的书写方法	13
1.4.1 程序中的注释	13
1.4.2 Java 语言中的注释	14
1.4.3 编写注释时要注意的问题	15
模块 1.5 UltraEdit 程序编辑工具的使用方法	15
1.5.1 配置 Java 运行环境	15
1.5.2 语法着色	16
1.5.3 【技能同步练习】	17
第2单元 Java 程序设计的基本技能	18
模块 2.1 Java 程序的组成	18
2.1.1 【案例 2-1】正方形类	18
2.1.2 【知识与技能】	22
2.1.3 【技能同步练习】	23
模块 2.2 基本数据类型及其运算	24
2.2.1 【案例 2-2】加法器	24
2.2.2 【知识与技能】	26
2.2.3 【技能同步练习】	32
模块 2.3 分支程序设计技术	32
2.3.1 【案例 2-3】数学软件包中符号函数与绝对值函数的编写	33
2.3.2 【知识与技能】	34
2.3.3 【技能同步练习】	36
模块 2.4 循环程序设计技术	36
2.4.1 【案例 2-4】求一个指定范围内的素数	36
2.4.2 【知识与技能】	38
2.4.3 【技能同步练习】	41

模块 2.5 程序设计中标识符的命名规范	41
第3单元 类的设计技术	43
模块 3.1 类的基本设计技能	43
3.1.1 【案例 3-1】设计电视机类	43
3.1.2 【知识与技能】	46
3.1.3 【技能同步练习】	50
模块 3.2 构造方法与静态成员设计技术	50
3.2.1 【案例 3-2】设计圆类	50
3.2.2 【知识与技能】	53
3.2.3 【技能同步练习】	58
模块 3.3 软件开发中类与方法的设计规则	59
3.3.1 类的设计规则	59
3.3.2 方法的设计规则	60
第4单元 复杂数据类型的使用技术	61
模块 4.1 字符串的使用技术	61
4.1.1 【案例 4-1】字符串的基本操作实例	61
4.1.2 【知识与技能】	62
4.1.3 【技能同步练习】	65
模块 4.2 数组的概念及其使用技术	66
4.2.1 【案例 4-2】求学生的平均成绩	66
4.2.2 【知识与技能】	67
4.2.3 【技能同步练习】	69
模块 4.3 向量的概念及其使用技术	70
4.3.1 【案例 4-3】向量的基本操作	70
4.3.2 【知识与技能】	72
4.3.3 【技能同步练习】	73
第5单元 软件的重用技术与接口的设计技术	74
模块 5.1 使用继承技术实现软件的重用	74
5.1.1 【案例 5-1】学生信息管理系统	74
5.1.2 【知识与技能】	79
5.1.3 【技能同步练习】	85
模块 5.2 抽象类的设计与使用技术	87
5.2.1 【案例 5-2】平面内的几何形状类	87
5.2.2 【知识与技能】	91
5.2.3 【技能同步练习】	97
模块 5.3 接口的设计与实现技术	97
5.3.1 【案例 5-3】动物类	97
5.3.2 【知识与技能】	100
5.3.3 【技能同步练习】	102
模块 5.4 对象与类的图形化表示	103

821	5.4.1 UML 简介	103
821	5.4.2 类的表示方法	103
821	5.4.3 接口的表示方法	104
	第6单元 程序中异常情况的处理技术	105
101	模块 6.1 提高程序健壮性的技术	105
101	6.1.1 【案例 6-1】程序中出现被除数是 0 的异常	105
101	6.1.2 【知识与技能】	106
101	6.1.3 【技能同步练习】	107
101	模块 6.2 Java 中异常处理程序的设计技术	107
101	6.2.1 【案例 6-2】对被除数是 0 的异常的处理	107
801	6.2.2 【知识与技能】	108
175	6.2.3 【技能同步练习】	111
175	模块 6.3 自定义异常类的设计技术	111
175	6.3.1 【案例 6-3】当年龄不在正常范围内时抛出自定义异常	111
175	6.3.2 【知识与技能】	113
175	6.3.3 【技能同步练习】	114
	第7单元 图形用户界面程序的设计技术	115
181	模块 7.1 应用程序窗口的设计技术	115
181	7.1.1 【案例 7-1】窗口示例	115
181	7.1.2 【知识与技能】	117
181	7.1.3 【技能同步练习】	120
181	模块 7.2 对话框的设计技术	120
181	7.2.1 【案例 7-2】数制转换器	121
181	7.2.2 【知识与技能】	123
181	7.2.3 【技能同步练习】	129
181	模块 7.3 窗口中常用组件的设计技术	129
181	7.3.1 【案例 7-3】用户信息选择窗口	129
181	7.3.2 【知识与技能】	132
181	7.3.3 【技能同步练习】	138
181	模块 7.4 交互式 GUI 程序的设计技术	138
181	7.4.1 【案例 7-4】水费计算程序	138
181	7.4.2 【知识与技能】	141
181	7.4.3 【技能同步练习】	144
181	模块 7.5 窗口中组件布局的设计技术	145
181	7.5.1 【案例 7-5】温度转换器	145
181	7.5.2 【知识与技能】	147
181	7.5.3 【技能同步练习】	150
181	模块 7.6 具有菜单的窗口设计技术	151
181	7.6.1 【案例 7-6】设计一个文本编辑器	151
181	7.6.2 【知识与技能】	155
181	7.6.3 【技能同步练习】	157

第8单元 文件访问技术	158
模块 8.1 操作系统文件的操作技术	158
8.1.1 【案例 8-1】文件与目录操作	158
8.1.2 【知识与技能】	160
8.1.3 【技能同步练习】	161
模块 8.2 字节型文件的访问技术	161
8.2.1 【案例 8-2】创建字节流文件并读取其内容	161
8.2.2 【知识与技能】	164
8.2.3 【技能同步练习】	167
模块 8.3 字符型文件的访问技术	167
8.3.1 【案例 8-3】文本编辑器	168
8.3.2 【知识与技能】	172
8.3.3 【技能同步练习】	174
模块 8.4 文件的随机访问技术	174
8.4.1 【案例 8-4】学生信息的读写	174
8.4.2 【知识与技能】	177
8.4.3 【技能同步练习】	179
第9单元 用 Java 程序访问数据库的技术	180
模块 9.1 查询数据表的技术	180
9.1.1 【案例 9-1】显示学生信息	180
9.1.2 【知识与技能】	182
9.1.3 【技能同步练习】	188
模块 9.2 修改数据表的技术	189
9.2.1 【案例 9-2】创建一个存放学生信息的数据表	189
9.2.2 【知识与技能】	191
9.2.3 【技能同步练习】	194
第10单元 多线程程序的设计技术	196
模块 10.1 如何创建与理解线程	196
10.1.1 【了解线程概念】	196
10.1.2 【案例 10-1】用 Thread 类创建多线程	196
10.1.3 【知识与技能】	198
10.1.4 【技能同步练习】	199
模块 10.2 使用 Runnable 接口创建多线程	199
10.2.1 【案例 10-2】实现 Runnable 接口创建多线程	199
10.2.2 【知识与技能】	200
10.2.3 【技能同步练习】	201
模块 10.3 两种实现多线程方式的对比分析	201
10.3.1 【案例 10-3】模拟铁路售票系统	201
10.3.2 【知识与技能】	204
10.3.3 【技能同步练习】	205
模块 10.4 线程的同步——同步代码块	205

10.4.1	【案例 10-4】模拟铁路售票系统	205
10.4.2	【知识与技能】	209
10.4.3	【技能同步练习】	210
模块 10.5	线程的同步——同步函数	210
10.5.1	【案例 10-5】模拟铁路售票系统	210
10.5.2	【知识与技能】	212
10.5.3	【技能同步练习】	212
模块 10.6	线程间的通信	212
10.6.1	【案例 10-6】生产者线程与消费者线程的通信问题	213
10.6.2	【知识与技能】	216
10.6.3	【技能同步练习】	216
参考文献		217

第1单元 程序运行环境的建立与程序的编译调试技术

本单元介绍 Java 语言的产生和发展历程, 阐述 Java 语言编写程序的主要优点, 引导读者构建 Java 开发环境; 通过实例介绍 Java 程序的两种方式: Java Application 和 Java Applet, 让读者掌握在 JDK 环境中程序的调试、编译及运行技术; 介绍了 Java 程序的结构及编写 Java 程序的常见问题, 旨在让未曾接触过 Java 编程语言的读者快速入门; 另外, 介绍了 Java 开发工具 UltraEdit 的使用方法及软件开发中程序注释的书写方法, 为后续各章的学习做好准备。

该单元学生要掌握的主要知识和技能如下。

- 了解 Java 语言的产生过程, 理解 Java 语言的主要优点。
- 掌握 Java 语言运行环境的建立与配置技术。
- 掌握在 JDK 环境中程序的调试、编译与运行技术。
- 初步掌握 UltraEdit 程序编辑工具的使用方法。
- 学会软件开发中程序注释的书写方法。

模块 1.1 Java 语言的产生

Java 是非常具有吸引力的面向对象编程语言, 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。应用 Java 更易于开发出高效、安全、稳定以及跨平台的应用程序。目前 Java 还处于快速发展阶段, 新的特性和应用仍在不断涌现。本模块对 Java 语言的产生以及发展进行简要介绍。

1.1.1 Java 语言的产生

(1) Java 的产生概述

随着 Java 技术的飞速发展, 越来越多的人加入到学习 Java 编程语言的热潮中。虽然现在 Java 无处不在, 但是其发展历史并不是很长。

20 世纪 90 年代初, 美国 Sun 公司的 Green 项目研究小组为了能够在消费电子产品上开发应用程序, 积极寻找合适的编程语言。消费电子产品种类繁多, 即使是同一类电子产品所采用的处理芯片和操作系统也不相同, 存在着平台不同的问题。当时最流行的编程语言是 C 语言和 C++ 语言, 但是, 对于消费电子产品而言, C++ 语言过于复杂和庞大, 并不适用, 安全性也并不令人满意。于是, 研究小组就着手设计和开发了一种名为 Oak (橡树) 的语言, 它是一种面向对象的、可移植的计算机语言。程序在执行前, 生成一个“中间代码”, 在任何一种机器上只要安装了特定的解释器, 就可以运行这个“中间代码”。这样的“中间代码”非常小, 解释器也不大, 这就是 Java 语言的前身。由于受家电科技的限制, 这个项目当时并没有成功。

在 Oak 就要夭折时, Internet 的迅速发展拯救了它, 1994 年, Green 小组将他们的开发转向了 Internet, 用 Oak 编写了一系列网络应用程序, 像网络浏览器 WebRunner 等。Oak 的新生让 Sun 公司看到了其发展的意义, 于是给 Oak 起了一个更响亮的名字——Java (此名源于印尼一个盛产咖啡的小岛——爪哇岛)。

1995 年, 随着 Java 在 Internet 和多媒体上的发展, 它得到了全球计算机业的高度重视。Netscape 和 Microsoft 公司都先后公布了支持 Java 语言的浏览器, 如 Netscape Navigator 2.0 以

上和 Microsoft Explore2.0 以上的版本。Sun 公司也推出了一种用 Java 编写、可嵌入到浏览器内部的小程序 Applet, 使得 Web 页面上出现了更丰富多彩的图像与动画。

Java 语言产生于 C++语言, 是一种完全的面向对象的编程语言, 其充分吸取了 C++语言的优点, 采用了程序员所熟悉的 C 语言和 C++语言的许多语法, 同时又去掉了 C 语言中指针、内存申请和释放等影响程序健壮性的部分, 可以说 Java 语言是站在 C++语言这个“巨人的肩膀上”前进的, 因此 Java 语言与 C++语言相比, 更加简洁、易于学习。

(2) Java 的发展

Java 最初的 1.0 版在设计上有不少缺点, Sun 在 1.2 版上倾注了大量的心血, 使其更加完善, 并加入了许多新的设计。鉴于 1.2 版本与之前版本的巨大差别, Sun 公司将 1.2 版及其以后版本命名为 Java 语言第二版 (Java2)。

在 2005 年之前的两三年时间里, Java2 的 1.4 版(J2SE1.4)被 IT 界广泛使用。

2004 年 9 月 30 日, J2SE1.5 发布, 对 J2SE1.4 做了许多技术和性能上的改进, 为了表示这个版本的重要性, J2SE1.5 更名为 J2SE5.0; 2005 年 6 月, JavaOne 大会召开, Sun 公司公布 Java SE6。此时 Sun 公司公布, Java 的各种版本已经更名, 取消了其中数字“2”(如 Java SE6 而不是 J2SE6)。

Java 现在的体系结构已经相当庞大, 按其应用不同, Sun 公司将 Java 分为三个版本。

① Java SE (即以前的 J2SE) Java 平台标准版 (Standard Edition), 主要是为台式机和 workstation 提供一个开发和运行的平台。本教程介绍的 Java 技术, 就是 J2SE。

② Java EE (即以前的 J2EE) Java 平台企业版 (Enterprise Edition), 建立在 Java SE 的基础上, 它是 JSP (Java Server Page)、Servlets、EJB、JTS (Java Transaction Service)、Java mail 以及 JMS (Java Message Service) 等多项技术的混合体。主要用于开发分布式、服务器端多层结构的应用系统, 如电子商务网站等。

③ Java ME (即以前的 J2ME) Java 平台小型版 (Micro Edition), 主要面向消费电子产品, 为消费电子产品提供一个 Java 的运行平台, 使得 Java 程序能够在手机、机顶盒、PDA 等产品上运行。

1.1.2 用 Java 语言编写程序的主要优点

Java 之所以成为当今比较流行的编程语言, 是由于它具有跨平台、面向对象、健壮且安全、多线程、简单易用等许多优越的特性。

(1) 平台无关性

学过 C/C++语言的都知道, 编写完 C/C++源代码后需要编译成机器码才能在机器上运行, 因此 C/C++是编译型的语言。还有一些语言完全是解释型的 (如 Basic), 不需要任何编译工作, 是边解释边执行的。与这些语言不同的是, Java 既是编译型的又是解释型的。在编写完 Java 源代码后, 首先要将其编译为一种中间状态的字节代码, 然后再由 Java 虚拟机 (JVM, Java Virtual Machine) 上的解释器来解释执行。这样, 任何一个系统, 只要具备了 Java 虚拟机, 那么它总是可以运行编译好的字节码, 如图 1-1 所示。

Java 源程序在编译后, 生成的中间代码叫字节码 (byte code), 字节码文件的扩展名为 .class, 只要安装了 Java 运行环境的机器都可以运行字节码程序, 因此 Java 实现了跨平台性, 即“一次编写, 处处运行”。

(2) 面向对象

和所有新一代的程序设计语言一样, Java 也采用了面向对象技术并更加彻底, 所有的 Java 程序和 Applet 程序均是对象, 面向对象技术具有封装性和继承性等特征。封装性实现了模块化和信息隐藏, 继承性实现了代码复用。并且 Java 语言为了提高编程效率, 提供了功能强大

的类库（或称为应用程序编程接口 API），可以帮助编程人员很容易地实现一些复杂的功能。面向对象技术使得应用程序开发变得简单，其编程过程更符合人们的思维方式。

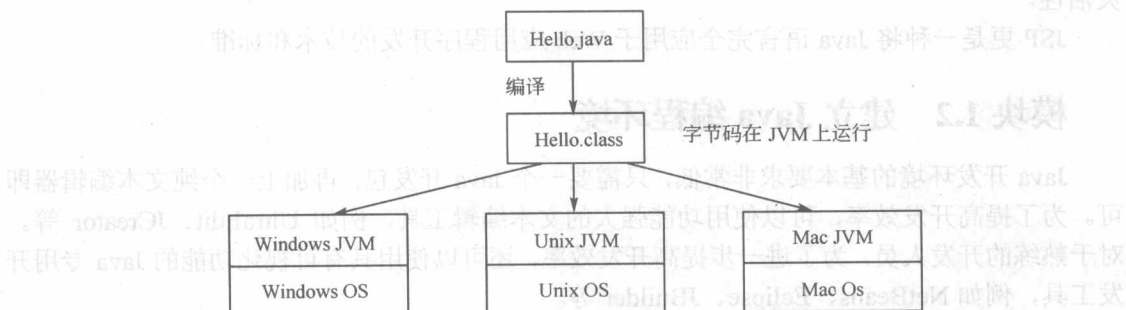


图 1-1 JVM 与跨平台性

(3) 简单性

设计 Java 语言的出发点就是容易编程，不需要深奥的知识。Java 语言的风格十分接近 C++ 语言，但要比 C++ 简单得多。Java 舍弃了一些不常用的、难以理解的、容易混淆的成分，如运算符重载、多继承等，增加了自动垃圾搜集功能，用于回收不再使用的内存区域。这不但使程序易于编写，而且大大减少了由于内存分配而引发的问题。

简单性还体现在小上。Java 解释器、系统模块和运行模块都比较小，适合在小型机器上运行，也适合从网上下载。

(4) 健壮性

每个程序设计人员都希望自己编写的程序更加可靠。在编写程序时，考虑周到一点固然能降低出错的可能性，但是如果语言本身就提供一系列的安全机制，就可以防患于未然，必能进一步提高程序的健壮性。Java 语言的设计目标之一，就是帮助程序员编写出高可靠性的程序，为此，Java 语言提供了很多技术用以提高程序的可靠性，如数组越界检查、运行时类型检查、取消指针操作以及无用单元自动回收等。

(5) 可移植性

Java 的可移植性，就是在不同的操作系统上源代码不做修改就能得到相同的运行结果。

同一数据类型在不同平台上的大小是不一样的，例如有的平台上整型（int）用 16 位二进制来表示，而有的平台上却是用 32 位二进制来表示。这样一个在用 32 位二进制表示整数的平台上开发的程序，拿到用 16 位二进制数表示整数的平台上去运行，很可能就会产生溢出问题。Java 中使用固定大小的数据类型解决了这个问题，例如整型始终用 32 位来表示，和底层平台的转换由 Java 虚拟机来完成。

可移植性问题不仅仅在于数据类型上，其他方面如图形界面的显示、多线程等都存在可移植性问题。

(6) 多线程

多线程机制使应用程序能够并行执行，而且同步机制保证了对共享数据的正确操作。通过使用多线程，程序设计者可以分别用不同的线程完成特定的行为，并使这些线程并行运行，这样就很容易地实现网络上的实时交互行为。Java 提供了完全意义上的多线程支持。

(7) 很好地支持网络编程

Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他对象。

Java 的 Applet 小应用程序可嵌入浏览器运行,能够完成动画、游戏、交互事务处理等功能,使 Web 页面具有动感,而不只是被动地展现文本和图形,提高了 Web 页面的智能性和灵活性。

JSP 更是一种将 Java 语言完全应用于 Web 应用程序开发的技术和标准。

模块 1.2 建立 Java 编程环境

Java 开发环境的基本要求非常低,只需要一个 Java 开发包,再加上一个纯文本编辑器即可。为了提高开发效率,可以使用功能强大的文本编辑工具,例如 UltraEdit、JCreator 等。对于熟练的开发人员,为了进一步提高开发效率,还可以使用具有可视化功能的 Java 专用开发工具,例如 NetBeans、Eclipse、JBuilder 等。

1.2.1 安装和设置 JDK

Sun 公司提供免费的 Java 开发工具包(JDK, Java Development Kit),可以从 Sun 官方网站上下载,也可以从其他途径获得,但是,最新的消息和版本必须从 Sun 的网站上才能够得到。

Sun 公司为不同的操作系统(如 Windows、Unix/Linux、Mac OS)提供了相应的 Java 开发安装程序。本书中使用 Windows 操作系统下的 Java 开发包。读者可以登录 Sun 公司的网站(<http://java.sun.com>),获取免费的 Java 开发包安装程序。本书中所给例题均以 J2SE5.0 版本为基础编写。

(1) Java 开发包的安装

在得到 Java 开发包后,首先需要进行安装。双击 Java 开发包安装程序,稍等片刻后,弹出如图 1-2 所示的许可证协议阅读画面。

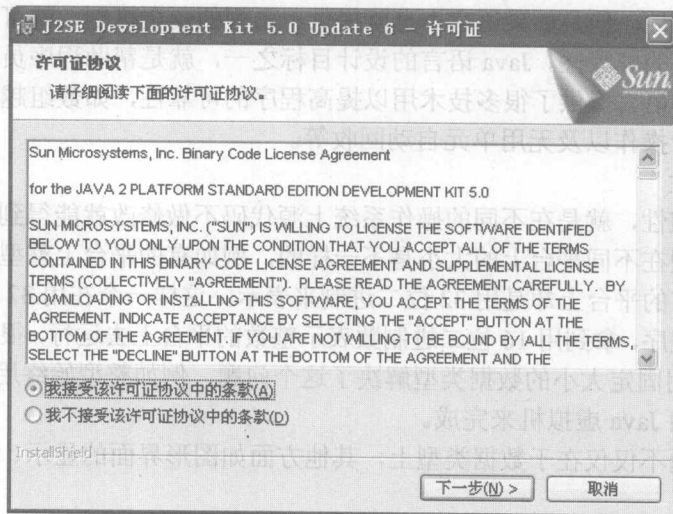


图 1-2 JDK 安装页面 1

选中接受条款后,单击“下一步”按钮继续安装。

在接下来弹出的对话框中,设定安装的组件和安装软件的路径。如果要重新设定安装的组件,可以单击图 1-3 所示组件名前的黑色小三角,打开下拉列表,选择是否安装,建议安装所有 Java 组件。如果要重新设定安装路径,可以单击图 1-3 所示的“更改”按钮,从弹出的对话框中选择文件的安装路径,可以在计算机的任何地方安装 JDK,但对于初学者来说最

好使用默认路径。

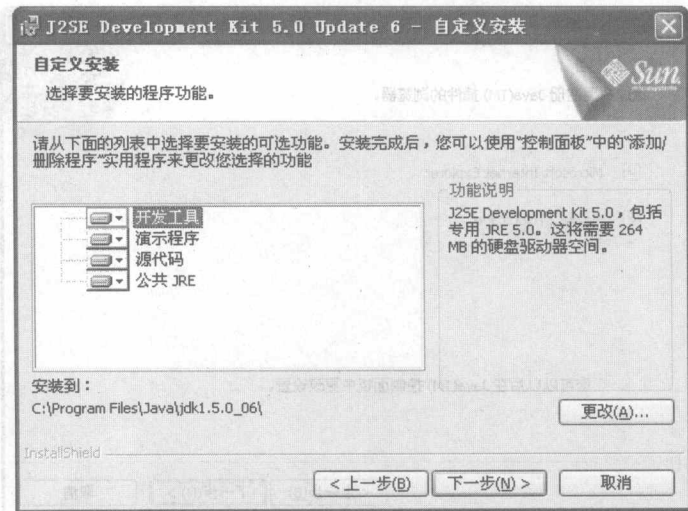


图 1-3 JDK 安装页面 2

点击图 1-3 所示的“下一步”按钮，开始安装 JDK 开发工具包，安装完成后，会启动 JDK 运行时安装向导，弹出如图 1-4 所示的安装窗口，可选择要安装的运行时组件，建议使用默认选择，单击“下一步”按钮继续。

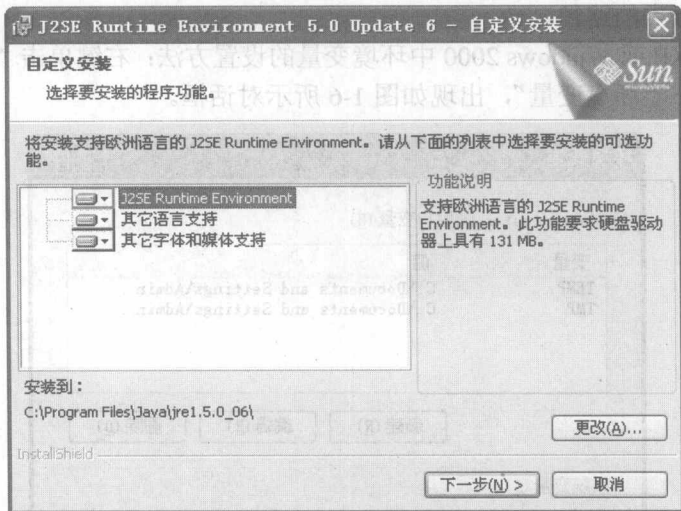


图 1-4 JDK 安装页面 3

此时，安装向导要求注册浏览器，如图 1-5 所示。单击“下一步”按钮继续，直到完成安装。

(2) 安装 Java 帮助文档

由于 JDK 的安装程序中并不包含帮助文档，因此帮助文档也必须从 Sun 公司的网站上下载进行安装。帮助文档通常安装在 JDK 所在目录的 docs 子目录下面。用浏览器打开 docs 子目录下的 index.html 文件，就可以阅读所有的帮助文档了。现在 Java 语言的一些爱好者将该帮助文档编译成了 CHM 格式的文档，使用起来非常方便，也可以从网络上下载 JDK 的 CHM

格式帮助文档。

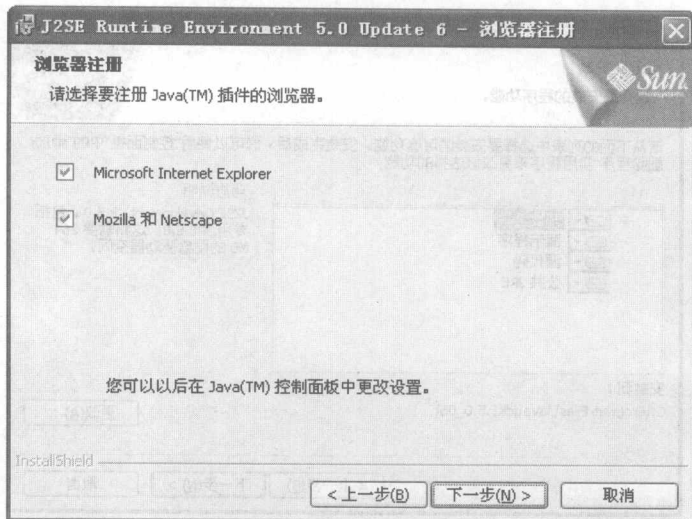


图 1-5 JDK 安装页面 4

(3) 配置环境变量

安装了 JDK 后，需要配置 path 和 classpath 两个环境变量。path 环境变量用来指定可执行文件（如 javac.exe、java.exe 等 JDK 工具）的搜索路径，而 classpath 则用来指定 JVM 在加载类文件时搜索类的路径。

在 Windows XP 或 Windows 2000 中环境变量的设置方法：右键单击“我的电脑”→“属性”→“高级”→“环境变量”，出现如图 1-6 所示对话框。

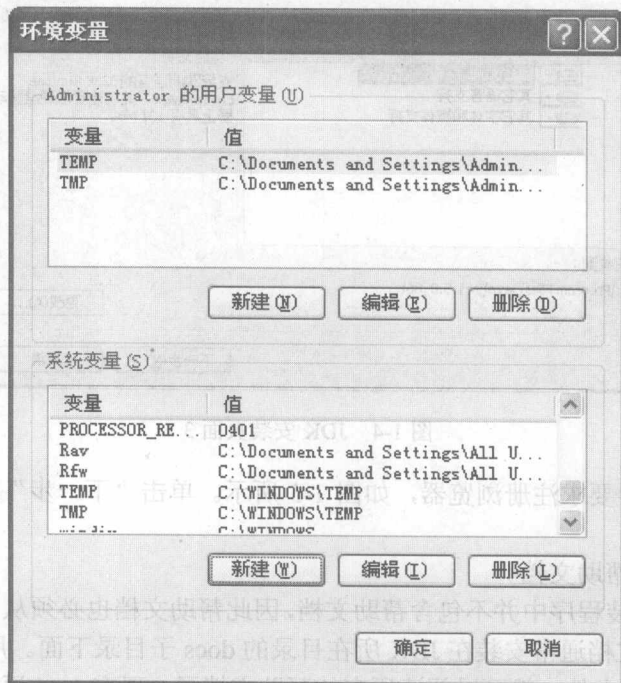


图 1-6 环境变量配置页面

在“系统变量”中点击“新建”按钮，弹出如图 1-7 所示对话框，变量名为“classpath”，变量值为“.;C:\Program Files\Java\jdk1.5.0_06\lib”。

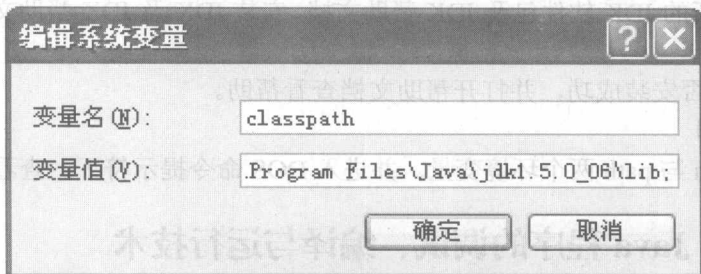


图 1-7 新建环境变量 classpath

其中，“.”代表当前目录，如果 classpath 中不包括该路径，则 JVM 并不到当前目录去搜索类文件；而“C:\Program Files\Java\jdk1.5.0_06\lib”则是 JDK 安装后系统类库所在的路径。

接下来，在“系统变量”列表中，选中“path”选项，单击“编辑”按钮，或双击“path”选项，打开“编辑系统变量”对话框，如图 1-8 所示。



图 1-8 编辑环境变量 path

在“变量值”文本框中，将光标移动到现有文本的最后，然后键入“.;C:\Program Files\Java\jdk1.5.0_06\bin”，其中，分号用来表示与前一个路径分开，“C:\Program Files\Java\jdk1.5.0_06\bin”代表 JDK 所在的目录 bin（JDK 工具对应的可执行文件所在的路径）。

设置好环境变量后，单击“开始”→“运行”输入“cmd”进入字符界面。在命令提示符界面中输入“javac -version”命令，此命令用于显示 Java 的版本信息，如图 1-9 所示。

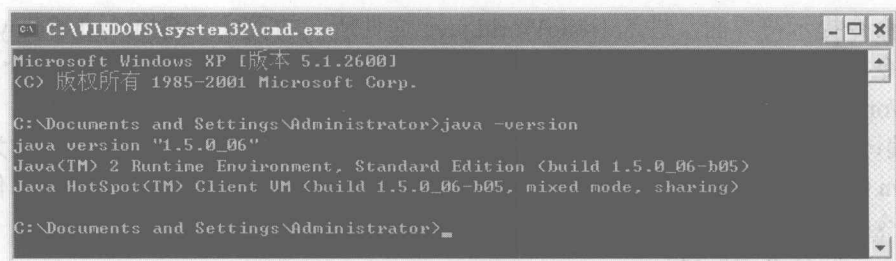


图 1-9 版本信息

然后，再试着在命令提示符下输入 javac 或 java 命令，如果有命令提示信息则表示 Java JDK 安装成功，系统变量更新。