



软件职业技术学院“十一五”规划教材

Java

高级程序设计

王路群 主 编 刘 嵩 刘 洁 副主编



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 高级程序设计

王路群 主 编

刘 嵩 刘 洁 副主编

中国水利水电出版社

内 容 提 要

Java 语言经过多年的发展,已经成长为优秀的主流开发语言。Java 语言具有面向对象、与平台无关、分布式、多线程等特点。尤其是 Java Swing 推出后,使得 Java 语言的功能更加强大。

本书通过实例讲解了如何使用 Java 语言开发数据库应用程序、多媒体应用程序、网络应用程序等。这些程序典型简洁,主要功能突出,所涉及的技术可以解决同类问题。全书分为两个部分共 13 章,其中第一部分为 Java 高级编程的核心知识,介绍了 Java 语言基础、异常、多线程、Java 图形用户界面、Java Applet、Java 输入输出流、JDBC 数据库编程与 Java 网络编程,第二部分为扩展知识,介绍了 Java 常用 API、Java 多媒体编程与 Java Bean。

本书理论结合实例,难度适中,不仅可以作为计算机及相关专业的教材,同时也适合作为广大编程爱好者的自学教材。

本书所配电子教案可以从中国水利水电出版社网站上免费下载,网址为:
<http://www.waterpub.com.cn/softdown/>。

图书在版编目(CIP)数据

Java 高级程序设计 / 王路群主编. —北京:中国水利水电出版社, 2006

(软件职业技术学院“十一五”规划教材)

ISBN 7-5084-3907-4

I. J... II. 王... 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(2006)第 089443 号

书 名	Java 高级程序设计
作 者	王路群 主 编 刘 嵩 刘 洁 副主编
出版 发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (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	787mm×1092mm 16 开本 19 印张 460 千字
版 次	2006 年 8 月第 1 版 2006 年 8 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性教学环节的比重,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

1995 年底, Java 程序设计语言闯入了 Internet 领域, 并迅速占据了显著地位。距离 Sun 公司第一次发布 Java 已经整整 11 年了, 11 年对于计算机飞速发展的进程来说不算短, 它足以淘汰掉许多技术, 也足以考验真正的强者。时至今日, Java 已成为 Internet 中最受欢迎、最具影响的编程语言之一。

本书通过一个项目实例, 引导读者进入神奇的 Java 世界, 通过这个案例程序, 本书展示了 Java 语言的主要技术特征, 在相关章节中介绍和分析了相关技术点在案例中的应用, 最后生成了一个用 Java 语言搭建的完整的应用程序。

本书包含两个部分共 13 章, 通过项目实例贯穿了第一部分的绝大部分关键技术。第 1 章介绍 Java 语言的特点和基本语法。第 2 章介绍 Java 异常处理的基本原则。第 3 章介绍了 Java Applet。第 4 章开始介绍 Java 的图形用户界面以及图形界面的事件处理机制。第 5 章介绍了 Java 中的事件处理机制。第 6 章介绍了图形用户界面 Swing 的用法。第 7 章介绍了 Java 的多线程机制。第 8 章介绍了 Java 中流传输的语法、技巧和应用。第 9 章介绍了 Java 中数据库的连接技巧与应用。第 10 章介绍了 Java 强大的网络编程。本书第二部分为补充知识, 第 11 章介绍了 Java 丰富的 API 使用。第 12 章介绍了在 Java 中如何运用多媒体技术制作生动的 Java 程序。第 13 章介绍了 Java Bean 的原理以及使用。

本书的最大特点就是以项目实例为基线, 贯穿 J2SE 中的关键技术。每章首先介绍 Java 关键技术点, 并在每章最后完善本书案例。在每一章的开始都有教学要求, 介绍本章的主要内容以及作者建议的学习重点, 引导读者有目的地学习, 达到事半功倍的效果。

本书由王路群任主编, 刘嵩、刘洁任副主编, 肖英、鲁立、谢日星、郭丽、胡轶萌、陈暄参加编写, 刘嵩、刘洁、肖英统编全稿。

由于时间仓促, 加之编者水平有限, 书中不妥或错误之处在所难免, 恳请广大读者批评指正。作者 E-mail: luqunwang@163.com。

编 者

2006 年 5 月

目 录

序

前言

网络聊天室项目需求.....	1
----------------	---

第一部分

第 1 章 Java 入门基础	2
1.1 Java 语言入门	2
1.1.1 Java 语言特点	2
1.1.2 Java 语言的运行环境	3
1.1.3 Java Application 程序与 Java Applet 程序	6
1.2 标识符、关键字和数据类型	9
1.2.1 标识符	9
1.2.2 关键字	9
1.2.3 分隔符	9
1.2.4 数据类型	10
1.2.5 常量与变量	12
1.3 运算符、表达式和语句	13
1.3.1 运算符	13
1.3.2 表达式	16
1.3.3 流程控制语句	17
1.4 类、对象和接口	23
1.4.1 类与对象	23
1.4.2 继承	26
1.4.3 包	28
1.4.4 接口	29
项目实训	30
第 2 章 异常	33
2.1 什么是异常	33
2.1.1 异常处理基础	33
2.1.2 Java 异常处理机制	33
2.2 异常的类型	34
2.2.1 Java 异常类类型	34

2.2.2	Java 异常类组成	35
2.3	未被捕获的异常	35
2.4	try...catch 语句	36
2.5	自定义异常与 throw 关键字	38
2.5.1	创建自己的异常	38
2.5.2	使用 throw 抛出异常	39
2.6	throws 与 finally 关键字	40
2.6.1	使用 throws 声明异常	40
2.6.2	使用 finally 关键字	41
2.7	Java 的内置异常	42
	相关知识链接	43
	程序引入异常的好处	43
	编程提示	44
	实例运用	44
	案例完善——聊天系统中的异常处理	45
	项目实训	46
第 3 章	Java Applet	47
3.1	HTML 简介	47
3.2	浏览器处理 Applet	48
3.3	Applet 类及其方法	49
3.3.1	Applet 类的层次	49
3.3.2	Applet 运行状态控制基本方法	50
3.3.3	Applet 绘图	51
	相关知识链接：关于 Applet 的一些细节	52
	实例运用	54
	项目实训	55
第 4 章	图形用户界面 GUI（一）——AWT	56
4.1	AWT 工具集简介	56
4.2	AWT 事件模型	57
4.3	GUI 组件上的图形操作	58
4.3.1	Graphics 类	59
4.3.2	组件重绘	60
4.3.3	绘制基本图形	60
4.3.4	建立字体	63
4.4	常用的 AWT 组件	64
4.4.1	文本框和文本区	65
4.4.2	按钮和标签	67

4.4.3	面板和画布	69
4.4.4	布局设计	71
4.4.5	选择型组件	77
4.4.6	Component 类的常用方法	80
4.4.7	Dialog 类	82
相关知识链接		86
关于 Java2D、Java3D 技术		86
项目实训		88
第 5 章	Java 中的事件处理机制	89
5.1	事件与事件处理	89
5.2	授权事件模型	90
5.3	事件类及其处理	92
5.3.1	ActionEvent 类	92
5.3.2	AdjustmentEvent 类	97
5.3.3	ComponentEvent 类	99
5.3.4	FocusEvent 类	99
5.3.5	ItemEvent 类	101
5.3.6	MouseEvent 类	102
5.3.7	KeyEvent 类	105
5.3.8	TextEvent 类	107
5.3.9	WindowEvent 类	108
5.4	事件适配器	110
5.5	事件处理的多重运用	111
相关知识链接		112
Inner 类		112
实例运用		113
案例完善——聊天系统中的事件处理		115
项目实训		116
第 6 章	图形用户界面 GUI (二) ——Swing	117
6.1	从 AWT 到 Swing	117
6.2	Swing 框架	118
6.3	JFrame、JDialog、JApplet	119
6.4	JPanel、JScrollPane、JSplitPane、JInternalFrame	128
6.5	图标和标签	134
6.6	各种组件	136
6.6.1	JButton	137
6.6.2	JCheckBox	137

6.6.3	JRadioButton	138
6.6.4	JComboBox	138
6.6.5	TextField、JTextArea、JPasswordField	138
6.6.6	JTextPane	144
6.6.7	JMenuBar、JMenu、JMenuItem、JToolBar	146
6.6.8	JTable	147
6.6.9	Timer、JProgressBar	152
6.6.10	JFileChooser	156
6.6.11	JTree	159
相关知识链接		161
paint(Graphics g)和 paintComponent(Graphics g)的区别		161
实例运用		162
项目实训		168
第 7 章	多线程	169
7.1	Java 线程模型	169
7.1.1	理解线程概念	169
7.1.2	用 Thread 类的子类创建线程	169
7.1.3	用 Runnable 接口创建线程	171
7.1.4	两种线程实现方式的对比分析	171
7.1.5	线程生命的控制	172
7.1.6	多线程优先级	173
7.2	Java 如何共享资源——线程同步	175
7.2.1	线程安全问题	175
7.2.2	使用线程同步	175
7.2.3	死锁问题	177
7.3	多线程之间的通信	177
实例运用		179
案例完善——聊天系统中的多线程应用		180
项目实训		181
第 8 章	I/O 输入输出流	182
8.1	流的概念	182
8.2	字节流、字符流和预定义流	182
8.2.1	字节流	182
8.2.2	字符流	183
8.2.3	预定义流	184
8.3	File 类	185
8.4	InputStream 和 OutputStream	188

8.4.1	FileInputStream	189
8.4.2	FileOutputStream	189
8.5	Reader 和 Writer	190
8.5.1	FileReader	191
8.5.2	FileWriter	191
8.6	RandomAccessFile 类	192
8.7	PipedInputSream 和 PipedOutputStream	194
8.8	过滤流与节点流	196
8.8.1	过滤流与节点流	196
8.8.2	BufferedInputStream 和 BufferedOutputStream	196
8.8.3	DataInputStream 和 DataOutputStream	197
	实例运用	198
	案例完善——聊天系统中的 I/O 流	203
	实例运用	204
第 9 章	Java 数据库连接 (JDBC)	205
9.1	JDBC 原理	205
9.1.1	JDBC 概述	205
9.1.2	JDBC 体系结构	206
9.2	JDBC 编程	207
9.2.1	JDBC 连接步骤	207
9.2.2	配置数据库	207
9.2.3	通过 executeQuery()方法进行查询	211
9.2.4	通过 executeUpdate()方法更新、添加、删除记录	212
	实例运用	214
	案例完善——聊天系统中的数据库操作	216
	项目实训	217
第 10 章	Java 网络编程	219
10.1	网络编程基础	219
10.1.1	统一资源定位器 (URL)	219
10.1.2	Internet 编址与端口号	219
10.1.3	客户/服务器模式	221
10.1.4	代理服务器	221
10.1.5	TCP/IP 与 UDP 协议	221
10.2	Java 和网络	222
10.2.1	网络类和接口	222
10.2.2	InetAddress 类	223
10.2.3	URL 类	223

10.2.4	URLConnection 类	224
10.3	Java 编写 TCP 网络程序	226
10.3.1	套接字概述	226
10.3.2	TCP/IP 客户套接字	226
10.3.3	TCP/IP 服务器套接字	227
10.4	Java 编写 UDP 网络程序	229
10.4.1	数据报服务器和客户	229
10.4.2	组播数据报	232
	实例运用	233
	案例完善——聊天系统中的网络编程	242
	项目实训	243

第二部分

第 11 章	Java API	244
	教学要求	244
11.1	理解 API 的概念——Java API 包	244
11.2	数据类型类	246
11.2.1	java.lang 包概述	246
11.2.2	Number 及其子类	246
11.3	数学运算工具——Math 类的方法	248
11.4	字符串处理	249
11.4.1	java.lang.String	249
11.4.2	java.lang.StringBuffer	251
11.5	java.lang 包	253
11.5.1	java.lang.Object	253
11.5.2	java.lang.Class	253
11.5.3	java.lang.System	254
11.5.4	java.lang.Runtime 与 java.lang.Process	254
11.5.5	java.lang.Package	254
11.6	java.util 包	255
11.6.1	Collection 接口	255
11.6.2	java.util.Random	255
11.6.3	java.util.Date	256
11.6.4	java.util.Calendar	257
11.6.5	java.util.StringTokenizer	258
	相关知识链接	260
	有效利用编程语言的帮助文档	260

项目实训	260
第 12 章 Java 与多媒体	262
12.1 Image 类	262
12.1.1 图像类型	262
12.1.2 创建图像	262
12.1.3 加载图像	263
12.1.4 显示图像	263
12.1.5 在应用程序中显示图像	265
12.1.6 ImageIcon 类	265
12.1.7 图像映射	266
12.2 利用 Java 播放动画	267
12.3 利用 Java 播放幻灯片	268
12.3.1 利用 Java 播放幻灯片	268
12.3.2 在 Applet 中播放声音	269
相关知识链接	271
Java 多媒体 API 参考	271
实例运用	271
项目实训	273
第 13 章 Java Bean 基础	275
13.1 理解 Java Bean	275
13.2 应用程序开发工具	276
13.3 Bean 开发工具包	277
13.4 JAR 文档格式	280
13.5 内省机制	281
13.6 开发一个简单的 Bean	281
13.6.1 如何创建自己的 Bean	281
13.6.2 创建一个简单的 Bean	281
13.6.3 测试 Bean	282
附录 1 Java 编码规范及文档注释	283
附录 2 Java2 平台介绍及专有名词注释	286
附录 3 Java IDE 工具介绍	288
参考文献	290

网络聊天室项目需求

设计内容

设计 C/S 模式的网络聊天室，客户端使用 Java Swing 应用程序与服务器的 Java 应用程序进行通信。

设计要求

- (1) 客户端在 Swing 应用程序服务器地址栏中输入服务器 IP 地址，与服务器程序通信。
- (2) 客户端在和服务器建立套接字连接后，用户输入聊天的昵称和密码提交给服务器验证，验证成功后可以进入聊天室。
- (3) 聊天室界面包含公共聊天信息和私人聊天信息，用户可以用鼠标单击聊天列表中某个聊天者的昵称，和该聊天者进行私人聊天。
- (4) 当新的用户进入聊天室时，所有的用户都可以在聊天列表中看到该用户的昵称；当某用户退出聊天室时，聊天列表将该用户的昵称删除，聊天室显示该用户离开的消息。
- (5) 用户可以注册新的用户名和密码。

总体设计

C/S 模式的网络聊天室程序共由 6 个 Java 源文件组成。Client.java、Chat.java、Reg.java、Login.java、SendMessage.java 五个 Java 文件生成的字节码打包后存放在客户机上，属于客户端程序；ServerThread.java 生成的字节码打包后存放在服务器上，属于服务器端程序。

(1) 客户端：

- Client.java 负责创建网络聊天室客户端主窗口。
- Login.java 负责生成用户输入昵称和密码的界面，用户单击“登录”按钮时，Client.java 将登录信息交给服务器，如果验证成功将进入聊天界面。
- SendMessage.java 和 Chat.java 共同组成了聊天界面。SendMessage.java 负责生成信息输入框和发送按钮，Chat.java 负责生成聊天信息显示界面和在线用户列表。
- Reg.java 负责生成用户注册界面。

(2) 服务器端：

- 负责监听用户的呼叫，为每个呼叫的用户启动一个线程，在该线程中和用户交流信息。如果用户提交了公共信息，就将该信息发给所有的用户。
- 如果用户提交了私人信息，就将该信息发给指定的用户。
- 如果用户提交登录信息，就对信息进行验证并判断用户是否已经登录过。
- 如果用户提交注册信息，就进行注册。
- 如果有新用户进入聊天室或者有用用户离开聊天室就提醒其他用户，将离开用户从列表中删除。

第一部分

第 1 章 Java 入门基础

教学要求

- 掌握典型的 Java 程序开发环境、Java 基本数据类型、输入输出语句及算术运算符、判定语句、面向对象的开发方法及句法结构。
- 理解 Java 语言的语言特性、面向对象的编程思想。
- 了解 Applet 与 Application 的区别、Java 在开发 Internet 和 Web 的分布式客户/服务器应用中的作用。

1.1 Java 语言入门

有人说，“以前，作为计算机软件人员，不能不会 C++，现在不能不会 Java”。这句话可能略失偏颇，但它确实道出了 Java 语言在软件界中越来越重要的地位。

Java 语言已经成为一种流行趋势，不管是计算机专业人员还是非计算机专业人员都应该接触和了解它，这样才能更好地适应不断发展变化的计算机技术环境。本章旨在让读者对 Java 语言有一个概念上的全面了解，至于 Java 的高级编程方面以及实用技巧，我们将在后续的章节陆续讨论。

好了，就让我们来进入这趟 Java 之旅吧！

1.1.1 Java 语言特点

Java 是目前使用最为广泛的网络编程语言之一，具有简单、动态、面向对象、分布式、解释执行、健壮、安全、结构中立、可移植、高效能、多线程等多种特点。

(1) 简单性。Java 语言是一种面向对象的语言，它通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。Java 略去了运算符重载、多重继承等模糊的概念，并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。另外，Java 也适合于在小型机上运行，它的基本解释器及类的支持只有 40KB 左右，标准类库和线程的支持也只有 215KB 左右。

(2) 面向对象。Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏，而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法，实现了代码的复用。

(3) 分布性。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。

(4) 鲁棒性。Java 在编译和运行程序时，都要对可能出现的问题进行检查，以避免错误的产生。它提供自动垃圾收集来进行内存管理，防止程序员在管理内存时产生错误。通过集成的面向对象的异常处理机制，在编译时，Java 提示出可能出现但未被处理的异常，协助程序员正确地进行选择以防止系统崩溃。另外，Java 在编译时还可捕获类型声明中的许多常见错误，防止动态运行时不匹配问题的出现。

(5) 安全性。用于网络、分布环境下的 Java 必须要防止病毒的入侵。Java 不支持指针，一切对内存的访问都必须通过对象的实例变量来实现，这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

(6) 体系结构中立。Java 解释器生成与体系结构无关的字节码指令，只要安装了 Java 运行时环境，Java 程序就可在任意的平台上运行。

(7) 可移植性。与平台无关的特性使 Java 程序可以方便地被移植到网络的不同机器上。同时，Java 的类库中也实现了与不同平台的接口，使这些类库可以移植。另外，Java 编译器是由 Java 语言实现的，Java 运行时环境由标准 C 实现，这使得 Java 系统本身也具有可移植性。

(8) 解释执行。Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时信息，使得连接过程更加简单。

(9) 高性能。和其他解释执行的语言，如 BASIC 不同，Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码，从而得到较高的性能。

(10) 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，我们可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，这样就很容易实现网络上的实时交互行为。

(11) 动态性。Java 语言的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。并且 Java 通过接口来支持多重继承，使之比严格的类继承具有更灵活的方式和扩展性。

1.1.2 Java 语言的运行环境

学习一门编程语言需要有一个程序开发环境。目前有许多优秀的软件产品提供了 Java 程序开发环境，我们在附录中对其进行了详细的介绍，读者若有兴趣，可自行查阅。但初学 Java 语言最好选用 Sun 公司推出的软件开发工具箱 JDK。

下面简单介绍一下 JDK 的下载和运行环境的设置，希望对初学者有所帮助。

(1) JDK 的下载。登录 Java 的官方网站 (<http://java.sun.com>) 免费下载 J2SDK，Java 官方网站界面如图 1-1 所示。

下载 J2SE 1.4.2 SDK 或 J2SE 5.0 均可，本节以 1.4.2 为例。单击对应的链接进入图 1-2 所示的下载界面。

在图 1-2 中，NetBeans 是 Sun 提供的 Java 开发环境；J2SE JRE 是 Java 的运行环境，如果你只需要运行现成的 Java 程序，而不需要编译 Java 源代码，则下载 JRE 即可；这里我们需要下载的是中间所标明的 JDK，单击该链接，即可看到新链接页面。

在新页面中首先需要选中 Accept License Agreement，然后根据操作系统平台选择合适的版本。另外还可以选择是全部下载后离线安装，还是在线安装。下载完成后，安装比较简单，直接双击，程序即可自动解压缩并进行安装，一路都有画面指示，只需要遵循即可。可自行选

择安装路径。安装完成后，在安装路径下的 bin 文件夹内有我们所需的各种开发工具。



图 1-1 Java 官方网站

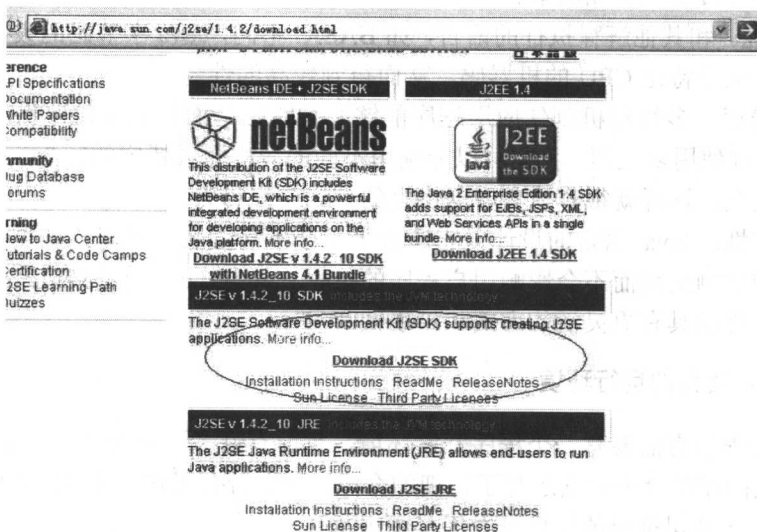


图 1-2 JDK 的下载页面

(2) 设置 Java 运行环境。进入“控制面板”，双击“系统”图标，弹出“系统属性”对话框，选择“高级”选项卡，单击“环境变量”按钮，弹出如图 1-3 所示的对话框。

单击“新建”按钮，添加如下的系统环境变量：

- 变量名：Path。
- 变量值：D:\jdk\bin（假设 JDK 安装在 D 盘根目录下）。

如果曾经设置过环境变量 Path，可选中该变量，然后单击“编辑”按钮，直接在弹出的如图 1-4 所示的对话框中对其进行修改即可。

但是，如果你的机器安装过一些商业化的 Java 开发产品或带有 Java 技术的一些产品，如 PB、Oracle 等，这些产品在安装后，可能会修改了 Path 的值，那么当运行编译器或者解释器