

21世纪
高职高专规划教材系列



Java语言

程序设计

赵国玲 王宏 柴大鹏 编著

第2版



免费提供电子教案

<http://www.cmpedu.com>



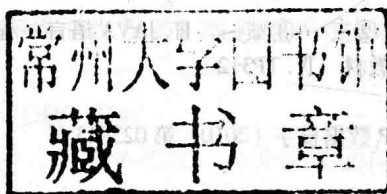
机械工业出版社
CHINA MACHINE PRESS

21 世纪高职高专规划教材系列

Java 语言程序设计

第 2 版

赵国玲 王宏 柴大鹏 编著



机械工业出版社

本书通过大量实例,深入浅出地介绍了 Java 语言的特点和基本语法、面向对象编程的基本概念和技术,以 Java 为载体介绍了面向对象程序设计的方法及对象、类、包、接口、继承等基本概念和实现方法,并介绍了 Java Applet、图形用户界面的创建与实现、事件的响应及处理、多线程、网络编程技术和数据库访问等。

本书概念清楚,层次分明,结构合理。书中列举的大量实例都具有较强的实用性,便于初学者模仿和借鉴。每章后均配有小结和习题,附录中还配有相应的实训练习。

本书既可以作为高职高专计算机专业的教材,也可作为广大 Java 爱好者的参考用书。

图书在版编目(CIP)数据

Java 语言程序设计/赵国玲,王宏,柴大鹏编著.—2 版.—北京:机械工业出版社,2010.3(2014.7重印)
(21世纪高职高专规划教材系列)
ISBN 978-7-111-29737-6

I. J… II. ①赵… ②王… ③柴… 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(2010)第 023913 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
责任编辑:鹿征
责任印制:刘岚

北京圣夫亚美印刷有限公司印刷

2014 年 7 月第 2 版·第 3 次印刷
184mm×260mm·19.75 印张·490 千字
6001—7800 册
标准书号:ISBN 978-7-111-29737-6
定价:32.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

封面防伪标均为盗版

读者购书热线:(010)88379203

前 言

Java 语言的诞生对 IT 产业带来了一次变革,从某种意义上讲对人们的日常生活也产生了深远影响。由于其平台无关性和与互联网发展的紧密结合,Java 语言已成为互联网和计算机应用的主流语言。目前,国内电子商务、金融、证券、电信等行业的大部分系统都正在或者准备采用 Java 技术来实现。Java 语言正逐步成为程序设计人员必须掌握的一种编程工具。

为了适应这一社会需求,Java 语言程序设计已成为高职高专计算机及相关专业的必修课程,旨在培养学生面向对象的编程思想,使其掌握使用 Java 语言进行程序设计的基本方法。为此,我们于 2004 年编写了《Java 语言程序设计》。但是,随着近几年 Java 技术的飞速发展和广泛应用,书中所使用的 JDK 版本已经落后,有些内容已经不能适应当前的实际应用,所以我们对《Java 语言程序设计》进行了修订。本书在保留原书特点的基础上,主要对以下内容进行了修改。

1) 本书更新了 JDK 版本,以 JDK 5.0 为蓝本。

2) 为了更好地培养学生面向对象的编程思想,突出面向对象程序设计这一主线,本书压缩了 Java 的基础语法,加强了面向对象编程思想的内容。

3) 本书加强了教材的实用性,每一个主题中都有一个实用性较强的实例。这些例子大多数是从实际程序开发中提取出来的,有较强的实用性和模仿性。最后还附有课程的实训指南。

4) 本书增加了网络和 JDBC 编程。

本书简要介绍了 Java 语言的基本语法,并介绍了面向对象程序设计的基本概念、特点和方法,在此基础上介绍了 Java 中各种面向对象技术的实现及各种实用程序的编写。通过学习本书,读者不仅可以掌握 Java 的语法知识,还可以理解面向对象程序设计的基本思想,掌握面向对象程序设计的方法,并为以后的进一步提高打下坚实的基础。

全书共分为 11 章:第 1 章介绍 Java 语言的特点及工作机制,并通过实例介绍 Java 程序的结构和 JDK 开发工具的使用;第 2 章主要介绍 Java 的基本语法知识,包括基本数据类型、常量变量的定义、运算符及数据的基本输入输出、数组及字符串;第 3 章主要介绍面向对象的基本概念及 Java 类、对象的建立和使用;第 4 章介绍继承、接口、抽象类等技术;第 5 章介绍 Java 中的异常类、异常处理机制以及异常的捕获、抛出等;第 6 章全面介绍 Java Swing 组件的创建、布局、使用及事件处理方法;第 7 章介绍 Java Applet 程序的创建和实现;第 8 章从线程的基本概念、实现机制、线程的状态和控制以及线程的同步 4 个方面对多线程进行介绍;第 9 章从输入输出流和文件与目录管理两个方面对 Java I/O 系统进行介绍;第 10 章介绍网络编程中的基本概念及一些网络编程类的使用;第 11 章对 JDBC 的基本概念、数据库的连接及访问进行详细的介绍。书中的重要章节都配有综合性较强的实例,每章都配有相应的小结和习题。附录中给出了与各章节对应的实训练习。

本书第 1 章和第 3~7 章由赵国玲编写,第 2 章由王宏编写,第 8~11 章由柴大鹏编写,附录由赵国玲、王宏、柴大鹏共同编写,全书由赵国玲统稿。为了提高实用性,编者在内容

和实例的选取上征求了山东鸥玛软件公司软件研发人员的意见，并进行了改进。

在本书的编写过程中还得到了山东电子职业技术学院各级领导和广大教师的大力支持和协助，在此表示由衷的感谢。

由于编者水平有限，书中难免存在错误和疏漏，恳请各位专家和读者批评指正。

为了配合教学，本书免费提供电子教案，读者可以从机械工业出版社教材服务网 <http://www.cmpedu.com> 下载。

编者

目 录

前言

第 1 章 认识 Java	1	2.4.3 逻辑运算符	30
1.1 Java 的发展及应用	1	2.4.4 赋值运算符	31
1.1.1 Java 的产生及发展	1	2.4.5 位运算符	32
1.1.2 Java 的广泛应用	2	2.4.6 其他运算符	35
1.2 Java 语言	2	2.4.7 运算符的优先级	35
1.2.1 Java 语言的特点	2	2.4.8 数值类型的转换	36
1.2.2 Java 的语言规范和运行机制	4	2.5 Java 程序的基本输入输出	37
1.3 Java 开发工具	5	2.5.1 基本输入	37
1.3.1 Sun JDK 软件包	6	2.5.2 基本输出	38
1.3.2 安装 JDK	8	2.6 流程控制语句	39
1.3.3 配置 JDK	8	2.6.1 选择语句	39
1.3.4 JDK 的常用工具	9	2.6.2 循环语句	43
1.3.5 使用 JDK 开发 Java 程序的一般 过程	10	2.6.3 跳转语句	46
1.3.6 几种常用的 Java 集成开发环境	11	2.6.4 应用实例	48
1.4 认识 Java 程序	12	2.7 数组	49
1.4.1 Java Application 程序	12	2.7.1 一维数组	49
1.4.2 Java Applet 程序	15	2.7.2 二维数组	52
1.5 Solaris 和 Linux 平台下 JDK 的 安装	18	2.7.3 应用实例	55
1.6 本章小结	21	2.8 字符串	56
1.7 习题	21	2.8.1 String 类	56
第 2 章 Java 编程基础	23	2.8.2 StringBuffer 类	62
2.1 标识符与关键字	23	2.9 本章小结	64
2.1.1 关键字	23	2.10 习题	64
2.1.2 标识符	24	第 3 章 面向对象概念及 Java 实现	70
2.1.3 分隔符	24	3.1 面向对象的基本概念	70
2.2 基本数据类型	25	3.2 在 Java 中使用类与对象	72
2.3 常量与变量	25	3.2.1 类的定义	72
2.3.1 常量	26	3.2.2 对象的创建	74
2.3.2 变量	27	3.2.3 对象的使用	76
2.4 Java 中的运算符与表达式	28	3.2.4 构造方法	78
2.4.1 算术运算符	29	3.2.5 对象创建实例	81
2.4.2 关系运算符	30	3.2.6 实例成员和类成员	82
		3.2.7 类及成员的访问权限	85
		3.3 方法的定义与实现	88

3.3.1	方法定义	88	5.2	异常处理	145
3.3.2	方法调用及参数传递	90	5.2.1	捕获异常	146
3.3.3	变量的作用域	94	5.2.2	声明异常	149
3.3.4	关键字 this 的使用	95	5.2.3	抛出异常	149
3.3.5	结束方法	96	5.3	自定义异常类	151
3.4	方法重载	96	5.4	本章小结	153
3.5	main()方法中的参数	99	5.5	习题	153
3.6	应用实例	100	第 6 章	Java 图形用户界面设计	155
3.7	本章小结	102	6.1	Java 图形用户界面概述	155
3.8	习题	102	6.1.1	AWT 与 Swing 组件	155
第 4 章	类的继承	105	6.1.2	Java 图形 API	156
4.1	类的继承	105	6.1.3	Java 图形用户界面的构成	157
4.1.1	继承的概念	105	6.1.4	图形用户界面程序实例	158
4.1.2	继承的实现	107	6.2	Java 布局管理	159
4.1.3	隐藏与覆盖	109	6.3	Java 事件处理	164
4.1.4	继承与构造方法	113	6.4	常用 Swing 组件的使用	167
4.1.5	关键字 super 的使用	115	6.4.1	JComponent 的常用属性和方法	167
4.1.6	阻止继承	117	6.4.2	容器组件	167
4.1.7	应用实例	117	6.4.3	按钮(JButton)	169
4.2	抽象类与抽象方法	120	6.4.4	标签(JLabel)	170
4.2.1	抽象方法	120	6.4.5	文本框(JTextField、JTextArea、 JPasswordField)	171
4.2.2	抽象类	121	6.4.6	复选框(JCheckBox)	174
4.3	接口的定义及实现	122	6.4.7	单选按钮(JRadioButton)	175
4.3.1	接口的定义	122	6.4.8	组合框(JComboBox)	177
4.3.2	接口的实现	123	6.4.9	列表框(JList)	178
4.3.3	应用实例	124	6.4.10	滚动条(JSlider)	180
4.4	包的应用	125	6.4.11	对话框(JOptionPane)	181
4.4.1	将类放入包	125	6.5	图形界面中的用户自定义成分	182
4.4.2	使用包中的类	127	6.5.1	绘制图形和文本	182
4.4.3	常见的系统包及类的使用	129	6.5.2	字体设置	185
4.5	内部类	136	6.5.3	颜色控制	186
4.6	应用实例	137	6.5.4	显示图像	187
4.7	本章小结	139	6.6	创建和使用菜单	187
4.8	习题	140	6.6.1	创建菜单	188
第 5 章	异常处理	141	6.6.2	弹出式菜单	190
5.1	异常和异常类	141	6.7	应用实例	190
5.1.1	什么是异常	141	6.8	本章小结	194
5.1.2	异常处理机制	143	6.9	习题	194
5.1.3	Java 异常类	144			

第7章 Java Applet 小应用程序	195	9.2.3 随机存取文件	238
7.1 Java Applet 的特点	195	9.3 应用实例	240
7.1.1 Java Applet 的执行过程	196	9.4 本章小结	241
7.1.2 Java Applet 的生命周期	197	9.5 习题	241
7.1.3 Applet 类的主要方法	198	第10章 Java 网络编程	244
7.2 HTML 基础	198	10.1 网络基础	244
7.2.1 HTML 常用标记	198	10.1.1 TCP/IP 协议	244
7.2.2 Applet 标记	199	10.1.2 通信端口	246
7.2.3 HTML 与 Applet 参数传递	200	10.1.3 URL 概念	247
7.3 应用实例	201	10.1.4 Java 与网络编程	248
7.4 本章小结	206	10.2 URL 编程	248
7.5 习题	206	10.2.1 创建 URL 对象	248
第8章 多线程	207	10.2.2 获取 URL 对象的属性	249
8.1 多线程的基本概念	207	10.2.3 使用 URL 类访问网络资源	250
8.2 多线程的实现机制	207	10.3 InetAddress 类	251
8.2.1 继承 Thread 类	207	10.4 Socket 通信	252
8.2.2 实现 Runnable 接口	208	10.4.1 Socket 类	252
8.2.3 程序实例	209	10.4.2 ServerSocket 类	254
8.3 控制线程和线程状态	210	10.5 应用实例	255
8.3.1 线程的控制	210	10.6 本章小结	258
8.3.2 线程的状态	214	10.7 习题	258
8.4 线程的同步	215	第11章 Java 对数据库的访问	259
8.4.1 共享受限资源	215	11.1 JDBC 数据库连接简介	259
8.4.2 线程间的协作	221	11.1.1 JDBC 的基本功能	259
8.4.3 死锁	222	11.1.2 JDBC 驱动程序类型	260
8.4.4 中断线程	222	11.2 结构化查询语言 SQL	261
8.5 应用实例	222	11.2.1 SQL 概述	261
8.6 本章小结	224	11.2.2 数据定义	261
8.7 习题	225	11.2.3 数据更新	262
第9章 Java 输入输出处理	227	11.2.4 数据查询	263
9.1 输入输出	227	11.3 JDBC 应用	264
9.1.1 输入输出概述	227	11.3.1 与数据库建立连接	264
9.1.2 字节流 InputStream 和 OutputStream	228	11.3.2 执行 SQL 语句	268
9.1.3 字符流 Reader 和 Writer	230	11.3.3 检索结果集	270
9.1.4 标准输入输出流	230	11.3.4 管理连接、语句和结果集	271
9.2 文件与目录	233	11.3.5 缓存结果集	271
9.2.1 Java 文件和目录管理	233	11.3.6 事务	272
9.2.2 文件的顺序访问	235	11.3.7 元数据	273
		11.4 应用实例	273

11.5 本章小结	279	实训 8 图形用户界面程序 设计 (1)	293
11.6 习题	279	实训 9 图形用户界面程序 设计 (2)	294
附录 全书实训	281	实训 10 Applet 小程序的设计	295
实训 1 Java 开发环境的安装、配置及 Java 程序运行	281	实训 11 多线程及其应用	297
实训 2 Java 基本语法	282	实训 12 Java 输入输出流	300
实训 3 数组和字符串的使用	286	实训 13 Java 网络编程	301
实训 4 面向对象的概念与 Java 实现	287	实训 14 Java 数据库访问	304
实训 5 类的继承	289	实训 15 Java 综合应用程序设计—— 网吧计费系统	306
实训 6 抽象类、包与接口的使用	289	参考文献	308
实训 7 异常处理	291		

第1章 认识 Java

Java 语言作为一种优秀的面向对象程序设计语言，目前已在网络编程中广泛使用。有专家预言，不久的将来，全世界 90% 的程序代码将用 Java 语言重写或改写。本章介绍 Java 的产生、发展和特点，并通过 Java Application 和 Applet 程序实例了解 Java 语言、认识 Java 程序的基本结构。

本章学习目标

- 了解 Java 及其发展历史
- 熟悉 Java 开发和运行环境
- 创建、编译、运行 Java 程序
- 认识 Java Application 程序
- 认识 Java Applet 程序
- 了解常用 Java 集成开发环境的使用

1.1 Java 的发展及应用

1.1.1 Java 的产生及发展

Java 是由 Sun 公司研发的进行网络、嵌入式项目开发的新一代程序设计语言。早在 1991 年，Sun 公司就由 Patrick Naughton 和 James Gosling 成立了 Green 项目组，为家用消费类电子产品开发分布式代码系统，解决家用电器的通信和控制问题。该项目组本来是以 C++ 语言作为设计软件的程序语言，但是他们随后发现 C++ 太复杂，而且安全性差。经过一番研究及通力合作，项目组最后基于 C++ 开发了一种新的语言。

Java 在它被设计成功之时名字叫 Oak，是因 Sun 公司玻璃窗外的一棵橡树而得名的。但后来设计小组发现 Oak 已被其他公司注册了。工程师们一边喝着咖啡，一边讨论取什么新名字。这种咖啡产于印尼爪哇岛。看着手上的咖啡，他们突然灵机一动，就叫它的英文名 Java 好了。这就是 Java 名称的由来。正因为此，Hot Java 的图标是一杯正冒着热气的咖啡。

1995 年春季，Sun 公司公布了完整的 Java 技术规范，其“Write Once, Run Anywhere”的口号使得 Java 一出现就得到了各 WWW 厂商的大力支持，他们纷纷在浏览器内部嵌入了用 Java 编写的 Applet(小应用程序)。Sun 和 Netscape 联合推出了开放式、与平台无关的对象脚本语言 JavaScript，使 Web 页面上出现了更丰富多彩的动画、图像。一时间“连 Internet，用 Java 编程”成为技术人员的一种时尚。

任何编程语言都离不开相应的开发工具和程序库，所以 Sun 公司于 1996 年发布了包括运行环境和开发工具在内的 JDK(Java Developer Kit) 1.0，之后又陆续发布新版本 JDK 1.1、JDK 1.2、Java 2 SDK(Software Development Kit)1.3、Java 2 SDK 1.4、Java 2 SDK 5.0 等。

从诞生到现在，Java 不断完善，并得到了广泛的应用。据伊文斯公司预计，北美地区使用 Java 的开发人员超过 61%。目前中国有 50% 以上的软件是使用 Java 开发的，而且持续保持

增长的势头。

说明: Sun公司于1998年12月发布了Java 2,它是应用Java最新技术的核心品牌。JDK1.2支持Java 2技术,自JDK1.2之后的JDK版本正式更名为Java 2 SDK。由于多数程序员已经习惯了名字JDK,所以本书中Java 2 SDK和JDK可以相互替换。

1.1.2 Java的广泛应用

Java已成为21世纪最重要和最具前途的网络编程语言,它使网络资源得到了最大限度地使用。Sun公司根据不同的计算机系统和用户的不同需求,开发了不同级别的Java工具软件。目前Java 2已发展出以下几个版本。

- J2EE: Enterprise Edition, Java 2 企业版,适用于企业开发的企业应用。
- J2SE: Standard Edition, Java 2 标准版,适用于个人开发的桌面应用。
- J2ME: Micro Edition, Java 2 微型版,适用于移动、手持设备开发的嵌入应用。
- Java Card: 适用于IC卡使用的版本,是为了用Java开发智能卡应用程序而制定的开发和运行环境标准。

所以,Java不仅可以用于网络程序的开发,也可以用来开发桌面应用系统和嵌入式系统等程序。概括起来,Java主要应用在以下几个方面。

- 1) 基于Internet的信息综合服务。
- 2) 基于Java芯片的智能卡及嵌入技术。
- 3) Java电子商务。
- 4) 可视化应用软件。

由此可见,学习、了解和使用Java语言已经成为时代的趋势。只有掌握并利用好Java语言才能跟上计算机技术发展的潮流。

1.2 Java语言

Java语言的迅速发展和广泛应用归功于它的设计和程序特征,特别是一次书写、多处运行更是其他程序设计语言无法比拟的。正如Sun公司在Java语言白皮书中所说,Java是简单的(Simple)、面向对象的(Object-oriented)、分布式的(Distributed)、解释型的(Interpreted)、健壮的(Robust)、安全的(Secure)、结构中立的(Architecture-neutral)、可移植的(Portable)、高效的(High-performance)、多线程的(Multithreaded)、动态的(Dynamic)。下面我们先简单了解一下这些概念,以便对Java语言有一个总体的认识。

1.2.1 Java语言的特点

1. 简单

实际上没有一门计算机语言是真正简单的,只是相比于其他面向对象程序设计语言,Java更简单一些。Java部分地模仿了C++,但是进行了极大的简化和改进,丢弃了C++中的指针,并用一种称为接口(Interface)的简单概念取代了C++的多重继承。Java采用自动分配和回收内存,而C++要求程序员去分配和回收内存。另外,清晰的语法使得Java程序容易编写和阅读。

2. 面向对象

早期的程序设计语言被认为是面向过程的 (Procedural Programming Language), 使用这种语言开发的软件系统是以过程实例为基础的, 程序设计注重的是解决问题的过程。而面向对象程序设计 (Object-oriented Programming, OOP) 是以对象为模型描述现实世界。任何事物都能抽象为对象, 一件物品、一个人、一个窗口图标, 甚至一张提货单据都可以认为是一个对象。

用 Java 编写程序主要进行对象创建、对象处理, 并使对象协调工作, 所以 Java 是面向对象的。

3. 分布式

分布式计算涉及几台计算机在网络上一一起工作。Java 一开始就设计了网络工作的能力, 从而使得分布式计算变得容易起来, 编写网络程序如同在文件中存取数据一样。如图 1-1 所示, 3 个程序运行在 3 个不同的系统上, 它们为完成 1 个共同的任务相互通信。

4. 解释型

运行 Java 程序需要一个解释器 (解释程序)。Java 程序被编译成 Java 虚拟机 (Java Virtual Machine, JVM) 编码, 这种编码称为字节码 (Byte Code)。如图 1-2 所示, 这种字节码是独立于计算机的, 它能在任何具有 Java 解释器的机器上运行。

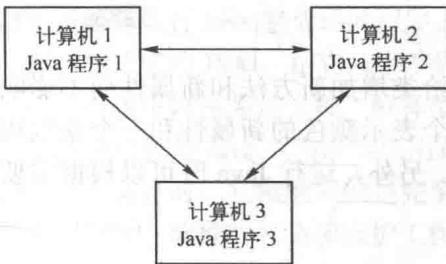


图 1-1 Java 程序在不同系统上共同工作

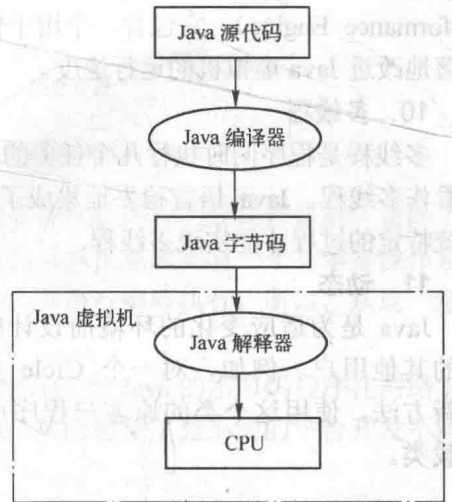


图 1-2 Java 解释器执行 Java 字节码

5. 健壮

“健壮的”是指可靠的 (Reliable)。任何一门程序设计语言都不能保证是绝对可靠的。Java 语言非常重视及早检查错误。Java 编译器可以查出许多其他语言运行时才能发现的错误。Java 语言丢弃了其他语言中容易引起错误的某些数据类型, 如指针等, 从而避免了因重写内存而造成数据破坏的可能性。

Java 具有实时异常处理功能, 有助于提高程序的可靠性。Java 强制程序员编写处理异常的代码, 所以它能够捕获并响应意外情况, 在发生错误时使程序正常执行错误处理代码, 从而使程序能够稳妥地结束。

6. 安全

作为 Internet 程序设计语言, Java 用于网络和分布式环境, 因此 Java 可以执行多层安全机制用以保护系统不受恶意程序破坏。

7. 体系结构中立

Java 引人注目的特点是体系结构中立, 也称与平台无关。利用 Java 虚拟机可以编写任何平台上都能运行的程序, 这成为 Java 应用软件便于移植的良好基础。

目前, Sun 公司为了适应不同的平台 (Windows、OS/2、Macintosh、各种 UNIX), 一种产品常常开发多个版本, 大多数操作系统上都安装了 Java 虚拟机。所以使用 Java 时, 开发者只需编写一个版本就能在各种平台上运行。

8. 可移植

Java 程序不必重新编译就能在任何平台上运行, 从而具有很强的可移植性。同时, Java 也没有平台特有的特征。有些语言的最大整数随平台不同而变化, 而 Java 中整数的范围在每个平台上都一样, 数值范围固定使得程序可以移植。

9. 高效

因为 Java 是解释型的, 字节码不在系统上直接运行, 而是通过解释器执行, 所以字节码的运行速度较之 C++ 之类的编译语言稍有逊色。但是随着 Sun 公司对 Java 技术的不断改进, Java 虚拟机的运行速度也不断提高。例如, Sun 公司开发了 Java HotSpot 运行引擎 (Java HotSpot Performance Engine)。它包含一个用于优化常用代码的编译器, 可以嵌入 Java 虚拟机, 从而显著地改进 Java 虚拟机的运行速度。

10. 多线程

多线程是程序同时执行几个任务的能力。例如, 一边下载视频文件一边播放录像就可以被看作多线程。Java 语言稳妥地集成了多线程程序设计技术, 而其他语言必须通过调用操作系统特定的过程才能实现多线程。

11. 动态

Java 是为适应变化的环境而设计的, 可以随便给类增加新方法和新属性而不影响该类的其他用户。例如, 对一个 Circle 类可以增加一个表示颜色的新属性和一个获取周长的新方法, 使用这个类的原客户程序可以照常运行。另外, 运行 Java 时可以根据需要来装载类。

1.2.2 Java 的语言规范和运行机制

1. Java 的语言规范

计算机语言都有严格的使用规则, 编写程序时必须遵守这些规则, 否则计算机不能编译执行。Java 语言规范是这门语言的技术定义, 包括它的语法、构成和预定义类的应用程序接口 (Application Programmer Interface, API)。Java 语言的语法和构成是相对稳定的, 而伴随着 JDK 新版本的不断推出, API 也一直在不断扩展。在 Java 软件的官方网站 (Java.Sun.com) 可以查看 Java 的语法规则、最新版本和 Java API 的更新。

Sun 公司使用称为 JDK 的 Java 开发工具箱发布 Java 的各个版本。JDK 是一个简单的命令行工具集, 包括软件库、编译 Java 源代码的编译器、执行 Java 字节码的解释器、测试 Java Applet 的浏览器, 以及其他实用工具。

2. Java 的运行机制

Java 语言之所以具有良好的跨平台性和可移植性，关键在于其独特的运行机制。在前面认识 Java 的特点时我们已经逐渐了解到这方面的一些内容，在此再对这一技术进行进一步归纳。

由 Java 语言编写的源程序代码经过编译后会生成一种二进制的中间代码，称为字节码，然后，通过运行与操作系统平台环境相对应的“Java 解释器”，字节码转化为特定系统平台下的机器码，最后系统解释执行这些代码。而其他语言则不同。例如，在 Windows 下编译一个 C++ 程序，编译器生成的可执行代码只能在 Windows 平台下运行。Java 程序与其他高级语言程序运行过程的区别如图 1-3 所示。

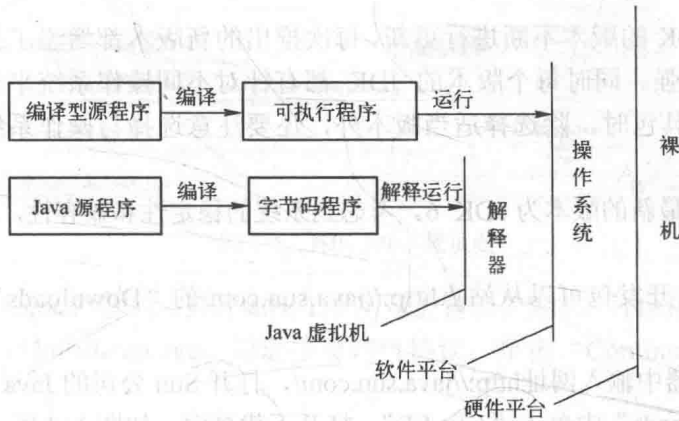


图 1-3 Java 程序与一般编译型程序的运行机制

Java 解释器又称为 Java 虚拟机 (JVM)，是驻留于计算机内存的逻辑计算机，实际上是一段负责解释执行 Java 字节码的程序。每个支持 Java 的计算机系统都有一个与本身操作系统和处理器相适应的 JVM，由它从字节码流中读取指令，并进行解释执行。所以，从这一意义上说，Java 也可以称为是一种“解释型”的高级语言。

Java 虚拟机是 Java 成为网络应用首选语言的秘密所在。当 Java 的字节码程序在网络上的不同机器上运行时，它接触到的是完全相同的解释器，从而避免了为不同的平台开发不同版本的应用程序，软件的升级和维护工作也大大简化。

1.3 Java 开发工具

“工欲善其事，必先利其器”。进行程序开发，首先要安装与程序开发相关的软件，并且熟悉这些工具软件的基本使用。随着 Java 语言的迅速发展，各大厂商纷纷推出了功能强大的开发工具。目前，Java 开发领域的开发工具呈现出百花齐放的局面，主要包括 Sun 公司的 JDK 软件包、Microsoft 公司的 Visual J++、Borland 公司的 Jbuilder、IBM 公司的 Visual Age for Java 和 WebSphere Studio、Oracle 公司的 Jdeveloper、WebGain 公司的 Visual Cafe、TogetherSoft 公司的 Together，以及几种小的编辑器（如 Jcreate、UltraEdit 等），还有开放源代码的 Eclipse、NetBeans 等。

本书介绍的是基于 Windows 平台的 Java 程序开发环境，使用的开发工具是 Java 2 JDK 5.0。

1.3.1 Sun JDK 软件包

Sun 公司不仅为 Java 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 JDK。开发人员和最终用户可以利用这个工具来开发 Java 程序。

JDK(Java Developer Kit)，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)，编程人员和最终用户可以利用这个工具来开发 Java 程序或调用 Java 内容。

Sun 公司对 JDK 的版本不断进行更新，每次推出的新版本都增添了很多新的类库并对原有的类进行了加强。同时每个版本的 JDK 都有针对不同操作系统平台的相应开发包，所以下载 JDK 工具包时，除选择适当版本外，还要注意选择与操作系统平台相适应的软件包。

目前，JDK 的最新的版本为 JDK 6。考虑到系统的稳定性和兼容性，本书以 JDK 5.0 为蓝本。

首次获取 JDK 开发包可以从站点 <http://java.sun.com/> 的“Downloads”下载，操作过程如下。

- 1) 在 IE 浏览器中输入网址 <http://java.sun.com/>，打开 Sun 公司的 Java 网站。
- 2) 在“Downloads”中单击“Java SE”，打开下载页面，如图 1-4 所示。

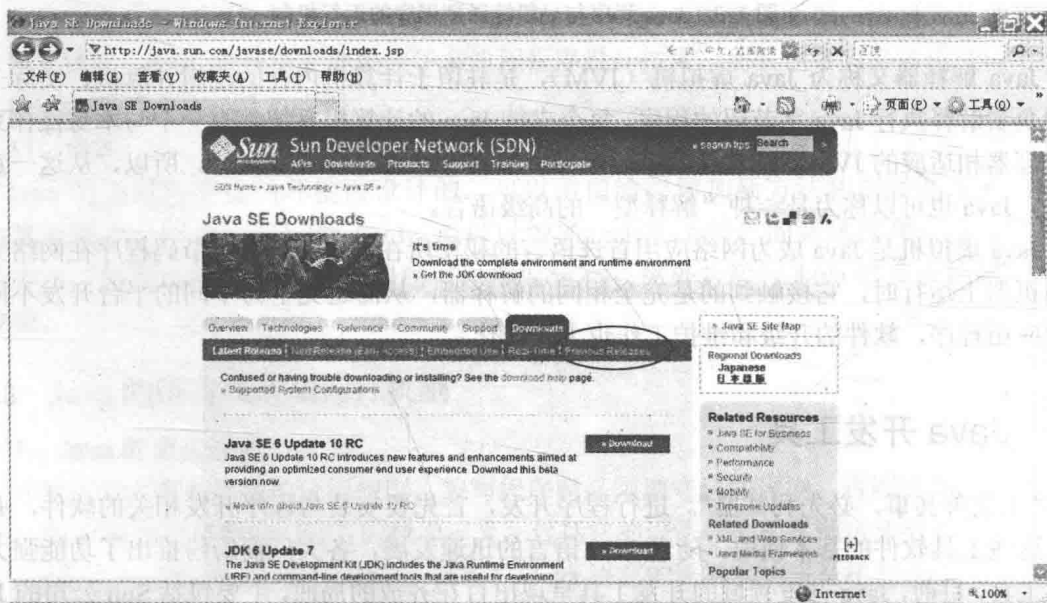


图 1-4 JDK 下载页面

3) 页面中显示的是当前最新的 JDK 版本。要下载 JDK 5.0，需要单击“Previous Releases”打开显示早期 JDK 版本的页面，选择其中的“J2SE 5.0 Downloads”，打开 JDK 5.0 的下载页面，如图 1-5 所示。

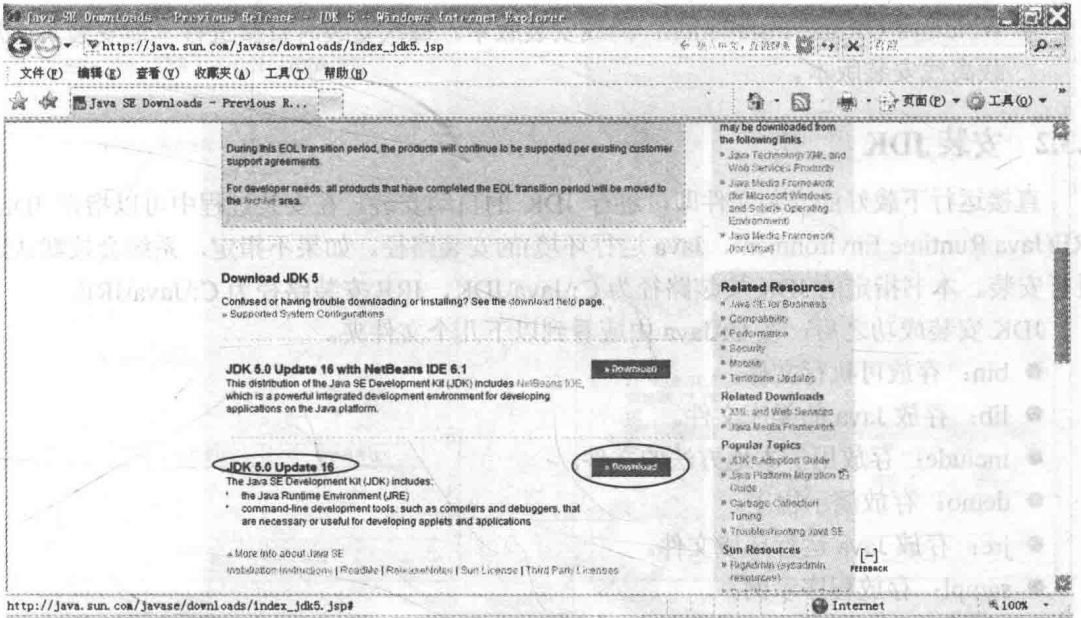


图 1-5 JDK 5.0 下载页面

4) 单击“Download”按钮，打开如图 1-6 所示的窗口。其中，“Platform”选择 Windows，“Language”选择 Multi-language，勾选接受许可协议，单击“Continue”按钮，打开下载连接页面。

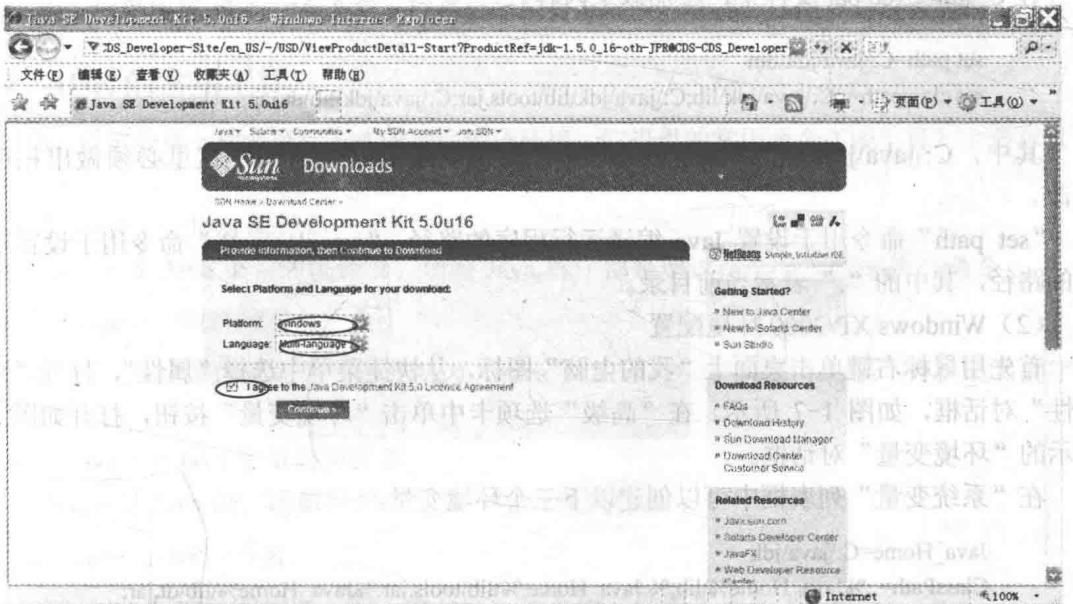


图 1-6 选择 JDK 平台和语言

5) 这里有以下两个下载连接。

- **Windows Online Installation:** 在线安装版本。每次安装时都从网上下载安装程序，下载完成后进行实际的安装。

- Windows Offline Installation: 离线安装版本。每次安装时直接进行本地安装。通常下载离线安装版本。

1.3.2 安装 JDK

直接运行下载好的安装文件即可进行 JDK 的自动安装。在安装过程中可以指定 JDK 和 JRE(Java Runtime Environment, Java 运行环境)的安装路径。如果不指定,系统会按默认路径进行安装。本书指定的 JDK 安装路径为 C:\Java\JDK, JRE 安装路径为 C:\Java\JRE。

JDK 安装成功之后,在 C:\Java 中应看到以下几个文件夹。

- bin: 存放可执行文件。
- lib: 存放 Java 的类库文件。
- include: 存放用于本地方法的文件。
- demo: 存放演示程序。
- jre: 存放 Java 运行环境文件。
- sampl: 存放程序示例。

1.3.3 配置 JDK

为了正确使用 JDK,需要手工配置一些环境变量。Windows 98 与 Windows XP/2000 的配置方法有所不同。

(1) Windows 98 环境配置

在 C:\autoexec.bat 文件中,添加以下内容。

```
set path=C:\java\jdk\bin
set classpath=.;C:\java\jdk\lib;C:\java\jdk\lib\tools.jar;C:\java\jdk\lib\dt.jar
```

其中, C:\java\jdk 是 JDK 的安装路径。如果安装的路径不同,这里必须做出相应的修改。

“set path”命令用于设置 Java 编译运行程序的路径,“set classpath”命令用于设置 Java 包的路径,其中的“.”表示当前目录。

(2) Windows XP/2000 环境配置

首先用鼠标右键单击桌面上“我的电脑”图标,从快捷菜单中选择“属性”,打开“系统属性”对话框,如图 1-7 所示。在“高级”选项卡中单击“环境变量”按钮,打开如图 1-8 所示的“环境变量”对话框。

在“系统变量”列表框中可以创建以下三个环境变量。

```
Java_Home=C:\java\jdk
ClassPath=.;%Java_Home%\lib;%Java_Home%\lib\tools.jar;%Java_Home%\lib\dt.jar;
Path=;%Java_Home%\bin
```

这是一种相对路径的设置方式,用%Java_Home%代替 JDK 的安装路径 C:\java\JDK。这样,当安装路径发生改变时,只需修改 Java_Home 的值,其他两个变量的值会随之改变。