



高等学校电子信息类专业规划教材

# 面向对象程序设计——Java

李金铭 林晓宇 宁正元 编著



清华大学出版社  
[Http://www.tup.tsinghua.edu.cn](http://www.tup.tsinghua.edu.cn)



北京交通大学出版社  
[Http://press.bjtu.edu.cn](http://press.bjtu.edu.cn)



21 世纪高等学校电子信息类专业规划教材

# 面向对象程序设计——Java

李金铭 林晓宇 宁正元 编著

清华大学出版社  
北京交通大学出版社  
· 北京 ·

## 内 容 简 介

本书旨在全面深入地介绍 Java 语言的特点及其面向对象的特性。

全书共分 15 章。书中的内容涵盖了 Java 语言的核心概念和知识。包括 Java 语言基础、面向对象编程基础、异常处理、图形化用户界面、输入/输出、多线程、Applet 以及 Java 语言的新特性——断言等。除了 Java 语言本身,本书有意增加了 Java 的应用部分。介绍了网络编程、J2EE 架构,还重点着墨于 JSP 动态网站开发技术。这样,读者不仅能从本书学到 Java 语言基础还能掌握实用的开发技术。通俗易懂的文字、典型翔实的例子将使 Java 不再令人生畏。读者将由浅入深逐步领略到 Java 世界的奥秘。

本书适合作为大中专院校计算机与相关专业的教材及自学 Java 语言的参考书,也可供计算机技术人员参阅。同时,本书涵盖了 Sun 公司 SCJP 认证的考点,可以作为 SCJP 认证的复习资料。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

面向对象程序设计:Java / 李金铭,林晓宇,宁正元编著. —北京:清华大学出版社;北京交通大学出版社,2005.7

(21 世纪高等学校电子信息类专业规划教材)

ISBN 7-81082-597-6

I. 面… II. ①李… ②林… ③宁… III. JAVA 语言-程序设计-高等学校-教材  
IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 091655 号

责任编辑:周益丹 特邀编辑:刘标

出版者:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>

北京交通大学出版社 邮编:100044 电话:010-51686414 <http://press.bjtu.edu.cn>

印刷者:北京鑫海金澳胶印有限公司

发行者:新华书店总店北京发行所

开 本:185×260 印张:18 字数:435 千字

版 次:2005 年 9 月第 1 版 2005 年 9 月第 1 次印刷

书 号:ISBN 7-81082-597-6/TP·220

印 数:1~5 000 册 定价:26.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043,51686008;传真:010-62225406;E-mail: [press@center.bjtu.edu.cn](mailto:press@center.bjtu.edu.cn)。

## 前 言

Java 语言是一门面向对象的语言。它摒弃了其他面向对象语言中存在的一些难于控制的技术和非面向对象特性,如多继承机制和独立于类之外的方法。因此,与其他面向对象语言如 C++ 相比,它显得更纯粹。从这点上讲,Java 语言是相对容易学习和掌握的。

Java 语言又是一门不断发展的语言。从 JDK1.0 开始,Java 的核心 API 已经历了数次变革,基于 Java 语言的各种技术已经扩展到信息领域的每个角落。应用 J2SE (Java 2 Platform Standard Edition),Java 可以用来开发桌面应用程序;应用 J2ME (Java 2 Platform Micro Edition),Java 可以开发嵌入式移动应用;应用 J2EE (Java 2 Platform Enterprise Edition),Java 可以用来开发企业级服务器应用程序。另外还有 Java Card 和 Java Web Services 等其他相关技术。这些技术发展迅速,也为 Java 语言的推广应用注入新的生命力。对于当今的程序员来说,不懂 Java 几乎寸步难行。

本书旨在全面而深入地介绍 Java 语言的特点及其面向对象的特性。同时,作者有意增加了 Java 的应用部分。介绍了网络编程、J2EE 架构,还重点着墨于应用 JSP 的 Web 开发技术。这样,读者不仅能从本书学到 Java 语言基础还能掌握实用的开发技术。

参与编写本书的人员有的具有丰富的 Java 教学经验,有的还拥有 Sun 公司的 Java 认证。作者认真研究了 Java 语言规范,编写了大量翔实的例子。每一个例子都有对应于章节的编号,便于读者检索。每一个例子都经过作者亲手调试,但要注意的是,并不是每个例子都可以运行。有一些特殊的实例可能出现编译期或者运行期错误,那是作者针对读者易犯的错误特意设置的,并且在书中都做了相应的解释。

全书共分 15 章。书中的内容包含了 Java 语言最新版本的核心概念和知识。同时涵盖 Sun 公司 SCJP 认证 1.2 和 1.4 版本的考点,所以又可以作为 SCJP 认证的复习指南。

本书第 1 章概述 Java 技术,介绍 Java 运行环境;第 2 章介绍 Java 语言的基础;第 3 章介绍用来控制程序流程的条件和循环语句;第 4 章引入了面向对象的知识,并介绍了用于面向对象的 UML 技术;第 5 章重点讲解面向对象的特性;第 6 章介绍 Java 中字符串的处理;第 7 章介绍异常处理机制,同时还介绍了 JDK1.4 版本中的断言机制;第 8 章介绍输入输出的处理;第 9 章介绍了一些实用的工具类:Wrapper 类、Math 类和容器类等;第 10 章介绍多线程编程;第 11 章介绍了图形用户界面的设计;第 12 章介绍 Applet;第 13 章介绍了网络通信编程;第 14 章全面介绍 J2EE 技术;第 15 章讲解了应用 Java 进行 Web 编程。

本书由宁正元执笔第 1 章,李金铭执笔第 2 章、第 3 章以及第 9 至 15 章,林晓宇执笔第 4 至 8 章。李金铭负责全书统稿。在本书的编写过程中,得到了许多同行的帮助,在此一并表示感谢。

由于作者水平有限,加之时间仓促,书中错漏之处在所难免,欢迎广大读者批评指正。

编 者  
2005 年 8 月

# 目 录

<b>第 1 章 Java 技术概述</b> .....	(1)
1.1 Java 的历史与现状 .....	(1)
1.1.1 Java 的出现 .....	(1)
1.1.2 Java 技术现状 .....	(1)
1.2 Java 工作原理 .....	(2)
1.3 Java 的特点 .....	(3)
1.3.1 简单、面向对象 .....	(3)
1.3.2 健壮并且安全 .....	(3)
1.3.3 结构中立并且可以移植 .....	(3)
1.3.4 高性能 .....	(4)
1.3.5 其他特性 .....	(4)
1.4 JDK 及 Java 运行环境 .....	(4)
1.4.1 安装 JDK .....	(4)
1.4.2 配置类路径 .....	(7)
1.4.3 第一个 Java Application .....	(8)
1.4.4 第一个 Java Applet .....	(8)
<b>第 2 章 Java 语言基础</b> .....	(10)
2.1 标识符与保留字 .....	(10)
2.1.1 标识符 .....	(10)
2.1.2 保留字 .....	(10)
2.2 变量 .....	(11)
2.2.1 变量的声明 .....	(11)
2.2.2 变量的使用 .....	(12)
2.3 基本数据类型 .....	(12)
2.3.1 整数类型 .....	(13)
2.3.2 浮点型 .....	(15)
2.3.3 字符型 .....	(16)
2.3.4 布尔型 .....	(18)
2.4 运算符 .....	(18)
2.4.1 算术运算符 .....	(18)
2.4.2 关系运算符 .....	(20)
2.4.3 布尔逻辑运算符 .....	(21)
2.4.4 位运算符 .....	(22)
2.4.5 赋值运算符 .....	(23)

2.4.6	条件运算符 .....	(23)
2.5	表达式 .....	(24)
2.5.1	自动类型转化 .....	(24)
2.5.2	强制类型转化 .....	(25)
2.5.3	运算符优先级 .....	(26)
2.6	基本数据类型数组 .....	(27)
2.6.1	数组的定义 .....	(27)
2.6.2	数组的初始化 .....	(28)
2.6.3	数组的使用 .....	(29)
<b>第3章</b>	<b>控制流程 .....</b>	<b>(31)</b>
3.1	分支语句 .....	(31)
3.1.1	条件语句 if-else .....	(31)
3.1.2	多分支语句 switch - case .....	(35)
3.2	循环语句 .....	(37)
3.2.1	while 语句 .....	(37)
3.2.2	do - while 语句 .....	(38)
3.2.3	for 语句 .....	(38)
3.3	跳转语句 .....	(39)
3.3.1	break 语句 .....	(40)
3.3.2	continue 语句 .....	(42)
3.3.3	return 语句 .....	(42)
3.4	注释语句 .....	(43)
<b>第4章</b>	<b>面向对象基础 .....</b>	<b>(44)</b>
4.1	面向对象基本概念 .....	(44)
4.1.1	什么是对象 .....	(44)
4.1.2	面向对象的方法 .....	(44)
4.2	类与对象 .....	(45)
4.2.1	类 .....	(45)
4.2.2	对象 .....	(45)
4.2.3	成员变量 .....	(46)
4.2.4	成员方法 .....	(48)
4.2.5	构造方法 .....	(52)
4.2.6	类成员与实例成员 .....	(54)
4.3	复合数据类型数组 .....	(57)
4.4	包 .....	(57)
4.4.1	package 语句 .....	(57)
4.4.2	包的编译 .....	(59)
4.4.3	import 语句 .....	(59)
4.4.4	包和 classpath 的设置 .....	(60)

4.5	UML 初步 .....	(61)
4.5.1	类图 .....	(63)
4.5.2	对象图 .....	(64)
4.5.3	序列图 .....	(65)
4.5.4	协作图 .....	(65)
4.5.5	状态图 .....	(66)
4.6	垃圾回收机制 .....	(67)
4.6.1	垃圾收集的意义 .....	(67)
4.6.2	命令行参数 .....	(67)
4.6.3	finalize 方法 .....	(68)
<b>第5章</b>	<b>面向对象特性 .....</b>	<b>(71)</b>
5.1	抽象性 .....	(71)
5.2	封装性 .....	(72)
5.2.1	private 限定词 .....	(73)
5.2.2	无限定词 .....	(74)
5.2.3	public 限定词 .....	(75)
5.3	继承性 .....	(76)
5.3.1	extends 关键字 .....	(76)
5.3.2	protected 限定词 .....	(77)
5.3.3	final 关键字 .....	(78)
5.4	多态性 .....	(79)
5.4.1	重载 .....	(79)
5.4.2	构造方法重载 .....	(81)
5.4.3	重写 .....	(82)
5.4.4	final 方法 .....	(85)
5.4.5	类型转换 .....	(85)
5.5	抽象类和接口 .....	(86)
5.5.1	抽象方法与抽象类 .....	(86)
5.5.2	接口 .....	(87)
5.6	内部类 .....	(89)
5.6.1	类成员内部类 .....	(89)
5.6.2	本地内部类 .....	(90)
5.6.3	静态内部类 .....	(91)
5.6.4	匿名内部类 .....	(92)
<b>第6章</b>	<b>字符串处理 .....</b>	<b>(93)</b>
6.1	String 类 .....	(93)
6.1.1	生成 String 对象 .....	(93)
6.1.2	获取字符串 .....	(94)
6.1.3	搜索字符串 .....	(98)

6.1.4	valueOf() 方法	(99)
6.2	StringBuffer 类	(99)
6.2.1	构造方法	(99)
6.2.2	长度和容量	(99)
6.2.3	修改字符串	(100)
6.3	字符串的比较	(101)
<b>第7章</b>	<b>异常处理和断言</b>	<b>(104)</b>
7.1	异常的概念	(104)
7.1.1	Java 异常处理机制	(104)
7.1.2	内置异常类	(105)
7.2	异常处理	(107)
7.2.1	捕获异常	(107)
7.2.2	声明抛弃异常	(110)
7.2.3	显式抛出异常	(111)
7.2.4	自定义异常	(111)
7.3	断言	(112)
7.3.1	assertion 的语法与语义	(112)
7.3.2	assertion 的编译	(113)
7.3.3	assertion 的运行	(113)
7.3.4	assertion 与继承	(115)
<b>第8章</b>	<b>输入/输出处理</b>	<b>(117)</b>
8.1	字节流	(117)
8.1.1	InputStream	(117)
8.1.2	OutputStream	(118)
8.2	文件处理	(118)
8.2.1	File 类	(118)
8.2.2	文件操作	(119)
8.2.3	目录操作	(121)
8.2.4	顺序流读写文件	(122)
8.2.5	随机读写文件	(125)
8.3	过滤流	(127)
8.3.1	DataInputStream 和 DataOutputStream	(128)
8.3.2	BufferedInputStream 和 BufferedOutputStream	(129)
8.3.3	PrintStream	(131)
8.4	字符流	(131)
8.4.1	Reader 和 Writer	(131)
8.4.2	InputStreamReader 和 OutputStreamReader	(132)
8.4.3	FileReader 和 FileWriter	(133)
8.4.4	BufferedReader 和 BufferedWriter	(134)



8.5	对象串行化 .....	(135)
8.6	序列流 SequenceInputStream .....	(137)
8.7	管道流 .....	(138)
<b>第9章</b>	<b>实用工具类 .....</b>	<b>(143)</b>
9.1	Wrapper 类 .....	(143)
9.1.1	构造函数 .....	(143)
9.1.2	Wrapper 类的应用 .....	(144)
9.2	Java. lang. Math 类 .....	(145)
9.3	容器类 .....	(146)
9.3.1	Arrays .....	(147)
9.3.2	LinkedList .....	(149)
9.3.3	Set .....	(152)
9.3.4	Map .....	(153)
<b>第10章</b>	<b>多线程 .....</b>	<b>(155)</b>
10.1	线程的概念 .....	(155)
10.2	线程的状态 .....	(156)
10.3	多线程编程 .....	(157)
10.3.1	创建线程 .....	(157)
10.3.2	线程的优先级和调度 .....	(159)
10.4	线程同步 .....	(161)
<b>第11章</b>	<b>图形用户界面设计 .....</b>	<b>(164)</b>
11.1	组件 .....	(164)
11.2	容器 .....	(164)
11.3	布局管理器 .....	(167)
11.3.1	FlowLayout 布局管理器 .....	(167)
11.3.2	BorderLayout 布局管理器 .....	(168)
11.3.3	CardLayout 布局管理器 .....	(170)
11.3.4	GridLayout 布局管理器 .....	(171)
11.3.5	GridBagLayout 布局管理器 .....	(172)
11.3.6	布局管理器的综合使用 .....	(173)
11.4	菜单 .....	(174)
11.5	AWT 组件库 .....	(176)
11.5.1	标签 .....	(176)
11.5.2	文本框和文本区 .....	(177)
11.5.3	按钮 .....	(179)
11.5.4	单、复选按钮和列表 .....	(181)
11.5.5	对话框和文件对话框 .....	(184)
11.6	事件处理机制 .....	(186)
11.6.1	事件监听与处理 .....	(187)

11.6.2	事件适配器 .....	(190)
<b>第 12 章</b>	<b>Applet .....</b>	<b>(192)</b>
12.1	Applet 的概念 .....	(192)
12.1.1	Applet 的运行 .....	(192)
12.1.2	Applet 与 Application 的异同 .....	(194)
12.1.3	Applet 的安全机制 .....	(194)
12.2	Applet 的生命周期及主要方法 .....	(194)
12.3	Applet 的绘制 .....	(196)
12.3.1	绘制控制方法 .....	(196)
12.3.2	图形绘制 .....	(198)
12.4	Applet 图像处理 .....	(205)
12.5	Applet 动画 .....	(207)
12.6	Applet 的声音技术 .....	(208)
12.7	JApplet .....	(209)
<b>第 13 章</b>	<b>网络通信 .....</b>	<b>(211)</b>
13.1	URL .....	(212)
13.2	Socket 通信 .....	(215)
13.2.1	Socket .....	(215)
13.2.2	Socket 通信的过程 .....	(216)
13.2.3	创建 Socket .....	(217)
13.2.4	客户端的 Socket .....	(217)
13.2.5	服务器端的 Socket .....	(218)
13.2.6	打开输入/输出流 .....	(218)
13.2.7	关闭 Socket .....	(219)
13.2.8	C/S 程序 .....	(219)
13.2.9	多客户的 C/S 程序 .....	(222)
13.3	数据报通信 .....	(225)
13.3.1	DatagramSocket 和 DatagramPacket .....	(225)
13.3.2	基于 UDP 的 C/S 程序 .....	(226)
13.3.3	用数据报进行广播通信 .....	(229)
<b>第 14 章</b>	<b>J2EE 技术 .....</b>	<b>(232)</b>
14.1	J2EE 的概念 .....	(232)
14.2	J2EE 的特点 .....	(233)
14.3	J2EE 的 4 层模型 .....	(234)
14.4	J2EE 的结构 .....	(236)
14.4.1	容器和服务 .....	(236)
14.4.2	容器类型 .....	(236)
14.5	J2EE 应用中的角色分配 .....	(237)
14.5.1	J2EE 产品提供商 .....	(238)

14.5.2	工具提供商 .....	(238)
14.5.3	应用程序组件开发者 .....	(238)
14.5.4	应用程序组装者 .....	(238)
14.5.5	应用程序部署者和系统管理员 .....	(238)
14.6	J2EE 的核心 API 与组件 .....	(239)
14.7	EJB .....	(240)
<b>第 15 章</b>	<b>JSP 编程</b> .....	<b>(242)</b>
15.1	Web 编程 .....	(242)
15.1.1	Web 的体系结构 .....	(242)
15.1.2	客户端向服务器端发送请求的两种方式 .....	(242)
15.1.3	Web 语言 .....	(243)
15.1.4	Tomcat 介绍及使用 .....	(244)
15.2	通用语法规则 .....	(248)
15.3	JSP 标签的分类 .....	(249)
15.4	JSP 指令 .....	(249)
15.4.1	page 指令 .....	(249)
15.4.2	include 指令 .....	(252)
15.5	脚本 .....	(252)
15.5.1	声明 .....	(252)
15.5.2	表达式 .....	(253)
15.5.3	小脚本 .....	(254)
15.6	注释 .....	(256)
15.6.1	内容注释 .....	(256)
15.6.2	JSP 注释 .....	(257)
15.6.3	脚本语言注释 .....	(257)
15.7	动作指令 .....	(257)
15.7.1	<jsp:param> .....	(258)
15.7.2	<jsp:include> .....	(258)
15.7.3	<jsp:forward> .....	(258)
15.8	内置对象 .....	(259)
15.8.1	Out 对象 .....	(259)
15.8.2	Request 对象 .....	(260)
15.8.3	Response 对象 .....	(260)
15.8.4	PageContext 对象 .....	(261)
15.8.5	Session 对象 .....	(261)
15.8.6	Application 对象 .....	(262)
15.8.7	Config 对象 .....	(262)
15.8.8	Page 对象 .....	(262)
15.8.9	Exception 对象 .....	(262)

15.9	表单初步 .....	(262)
15.10	JDBC 数据库编程 .....	(263)
15.10.1	从一个实例入手 .....	(263)
15.10.2	加载驱动程序 .....	(266)
15.10.3	连结数据库 .....	(267)
15.10.4	操作数据库 .....	(268)
15.10.5	关闭数据库连接 .....	(270)
15.10.6	对结果集的操作 .....	(270)
15.10.7	应用 JavaBean 和 JSP 连接数据库 .....	(271)
15.10.8	数据库连接池 .....	(273)
<b>参考文献</b>	.....	<b>(275)</b>

# 第 1 章 Java 技术概述

## 1.1 Java 的历史与现状

### 1.1.1 Java 的出现

目前,基于 Java 的各项技术已经成为 IT 业界的领先技术。尽管 Sun 公司设计师们的初衷只是想把 Java 应用在机顶盒、微波炉这样的智能家电设备上——他们想设计出一种可以跨不同硬件平台上通用的控制语言,但是正是他们的这种跨平台的思想造就了 Java 今天的成功。

互联网的出现使得计算模式进入了网络计算时代。网络计算模式的一个特点是计算机是异构的,即计算机的体系结构和操作系统是不一样的。例如 PC 的硬件是 INTEL 体系,操作系统是 Windows 或者是 Linux,而 Sun 工作站的硬件是 SPARC 体系,软件是基于 UNIX 的 Solaris 操作系统,相应的编程语言基本上只能应用在某种特定的平台上。网络计算模式的另一个特点是代码可以通过网络在各种计算机上进行迁移,这就迫切需要一种跨平台的编程语言,使得用它编写的程序能够在网络中各类计算机上正常运行。

1995 年,美国 Sun Microsystem 公司正式推出了 Java 语言,该语言具有安全、跨平台、面向对象、简单、适用于网络等显著特点。Java 语言的这些特点恰好符合互联网网络计算的要求,于是 Java 和 Internet 迅速融合并互相推动快速发展。IBM、Oracle、Microsoft、Netscape、Apple 等大公司纷纷与 Sun Microsystems 公司签订合同,授权使用 Java 平台技术。微软公司总裁比尔·盖茨先生曾经这样评价它:Java 语言是长时间以来最卓越的程序设计语言。

### 1.1.2 Java 技术现状

目前 Sun 公司把 Java 平台划分成 J2EE、J2SE、J2ME 共 3 个平台,针对不同的市场目标和设备进行定位。

J2SE(Java 2 Standard Edition)主要目的是为台式机和 workstation 提供一个开发和运行的平台。我们在学习 Java 的过程中,主要是采用 J2SE 来进行开发。

J2ME(Java 2 Micro Edition)主要是面向消费电子产品,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。

J2EE(Java 2 Enterprise Edition)主要目的是为企业计算提供一个应用服务器的运行和开发平台。J2EE 本身是一个开放的标准,任何软件厂商都可以推出自己的符合 J2EE 标准的产品,使用户可以有多种选择。IBM、Oracle、BEA、HP 等 29 家公司已经推出了自己的产品,其中尤以 BEA 公司的 weblogic 产品和 IBM 公司的 Websphere 最为著名。J2EE 将逐步发展成为可以与微软的 .NET 战略相对抗的网络计算平台。

上述 3 个 Java 平台的关系如图 1-1 所示。

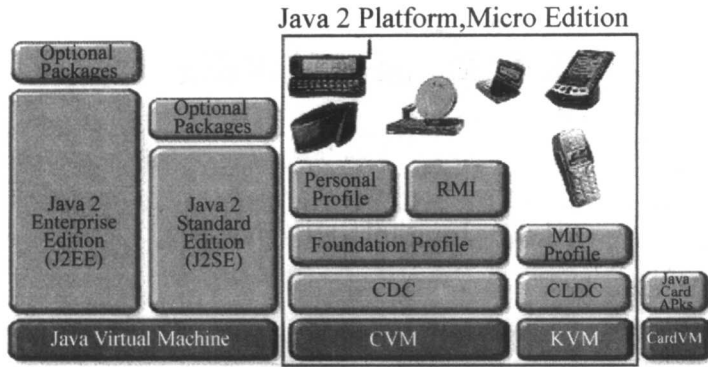


图 1-1 Java 平台

## 1.2 Java 工作原理

Java 语言是运行在 Java 虚拟机上的。Java 虚拟机是软件模拟的计算机,可以在任何处理器上(无论是在计算机中还是在其他电子设备中)安全并且兼容地执行 Java 应用程序。Java 虚拟机在计算机系统中的层次位置如图 1-2 所示。



图 1-2 Java 虚拟机在计算机系统层次位置

Java 编译器把 Java 程序源代码编译成 .class 文件,有时也可以称之为字节码文件。Java 程序的跨平台主要是指字节码文件可以在任何具有 Java 虚拟机的计算机或者电子设备上运行,Java 虚拟机中的 Java 解释器负责将字节码文件解释成为特定的机器码来运行。Java 程序的编译和执行过程如图 1-3 所示。

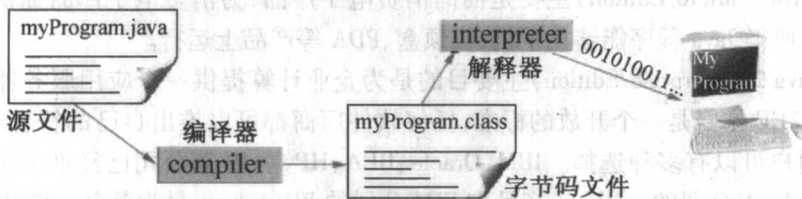


图 1-3 Java 程序编译执行过程

## 1.3 Java 的特点

### 1.3.1 简单、面向对象

Java 的简单首先体现在精简的系统上,力图用最小的系统实现足够多的功能;对硬件的要求不高,在小型的计算机上便可以良好地运行。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,封装性实现了模块化和信息隐藏,继承性实现了代码的复用,用户可以建立自己的类库。而且 Java 采用的是相对简单的面向对象技术,去掉了运算符重载、多继承的复杂概念,而采用了单一继承、类强制转换、多线程、引用(非指针)等方式。无用内存自动回收机制也使得程序员不必费心管理内存,使程序设计更加简单,同时大大减少了出错的可能。Java 语言采用了 C 语言中的大部分语法,熟悉 C 语言的程序员会发现 Java 语言在语法上与 C 语言极其相似。

### 1.3.2 健壮并且安全

Java 语言在编译及运行程序时,都要进行严格的检查。作为一种强制类型语言,Java 在编译和连接时都进行大量的类型检查,防止不匹配问题的发生。如果引用一个非法类型或执行一个非法类型操作,Java 将在解释时指出该错误。在 Java 程序中不能采用地址计算的方法通过指针访问内存单元,大大减少了错误发生的可能性;而且 Java 的数组并非用指针实现,这样就可以在检查中避免数组越界的发生。垃圾自动回收机制也增加了 Java 的健壮性。

作为网络语言,Java 必须提供足够的安全保障,并且要防止病毒的侵袭。Java 在运行应用程序时,严格检查其访问数据的权限,比如不允许网络上的应用程序修改本地的数据。下载到用户计算机中的字节代码在其被执行前要经过一个核实工具,一旦字节代码被核实,便由 Java 解释器来执行,该解释器通过阻止对内存的直接访问来进一步提高 Java 的安全性。同时 Java 极高的健壮性也增强了 Java 的安全性。

### 1.3.3 结构中立并且可以移植

网络上充满了各种不同类型的计算机和操作系统,为使 Java 程序能在网络的任何地方运行,Java 编译器编译生成了与体系结构无关的字节码结构文件格式。任何种类的计算机只有在其处理器和操作系统上有 Java 运行的环境,字节码文件就可以在该计算机上运行。即使是在单一系统的计算机上,结构中立也有非常大的作用。随着处理器结构的不断发展变化,程序员不得不编写各种版本的程序以在不同的处理器上运行,这使得开发出能够在所有平台上工作的软件集合是不可能的。而使用 Java 将使同一版本的应用程序可以运行在所有的平台上。

体系结构的中立也使得 Java 系统具有可移植性。Java 运行时系统可以移植到不同的处理器和操作系统上,Java 的编译器是由 Java 语言实现的,解释器是由 Java 语言和标准 C

语言实现的,因此可以较为方便地进行移植工作。

### 1.3.4 高性能

虽然 Java 是解释执行的,但它仍然具有非常高的性能,在一些特定的 CPU 上,Java 字节码可以快速地转换为机器码来执行。而且 Java 字节码格式的设计就是针对机器码的转换,实际转换时相当简便,自动的寄存器分配与编译器对字节码的一些优化可使之生成高质量的代码。随着 Java 虚拟机的改进和“即时编译”(just in time)技术的出现使得 Java 的执行速度有了更大的提高。

### 1.3.5 其他特性

为了易于实现跨平台,Java 被设计成为解释执行方式,字节码本身包含了许多编译时生成的信息,使连接过程更加简单。而多线程使应用程序可以同时进行不同的操作,处理不同的事件。在多线程机制中,不同的线程处理不同的任务,互不干涉,不会由于某一任务处于等待状态而影响了其他任务的执行,这样就可以很容易地实现网络上的实时交互操作。Java 在执行过程中,可以动态地加载各种类库,这一特点使之非常适合于网络运行,同时也非常有利于软件的开发,即使是更新类库也不必重新编译使用这一类库的应用程序。

## 1.4 JDK 及 Java 运行环境

Java 不仅是编程语言,还是一个开发平台,Java 技术给程序员提供了许多工具:编译器、解释器、文档生成器和文件打包工具等。同时 Java 还是一个程序发布平台,有两种主要的发布环境:首先是 Java 运行时环境(Java Runtime Environment, JRE),它包含了完整的类文件包,其次是许多主要的浏览器都提供的 Java 解释器和运行时环境。

Java Develop Kit 简称 JDK,是 Sun 公司为全世界的 Java 程序员提供的一个免费的 Java 程序开发包和文档。JDK 包括了 Java 编译器命令“Javac”以及 Java 程序执行命令“Java”,还有帮助文档生成器命令“Javadoc”等。所有这些命令都可以在命令行下运行。除了这些命令,JDK 还包含了 Java 运行环境以及可以供用户调用的 API(应用程序接口)。JDK 和帮助文档可以在 Sun 公司的网站下载:<http://java.Sun.com>。

### 1.4.1 安装 JDK

下面以 JDK1.4.2 为例介绍如何安装 JDK 开发包。运行从 Sun 公司网站下载的 j2sdk-1\_4\_2\_07-windows-i586-p.exe,开始解压这个安装包,出现的画面如图 1-4 所示。在下面的步骤里,我们可以接受 Sun 公司的许可条约,选择安装目录和安装组件以及是否为系统现有的浏览器安装 Java 插件等(图 1-5 至图 1-9)。



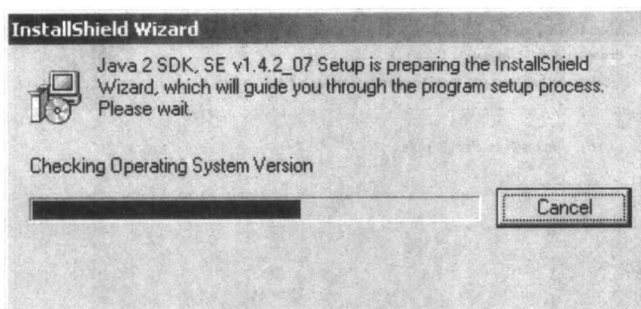


图 1-4 解压 JDK 安装包

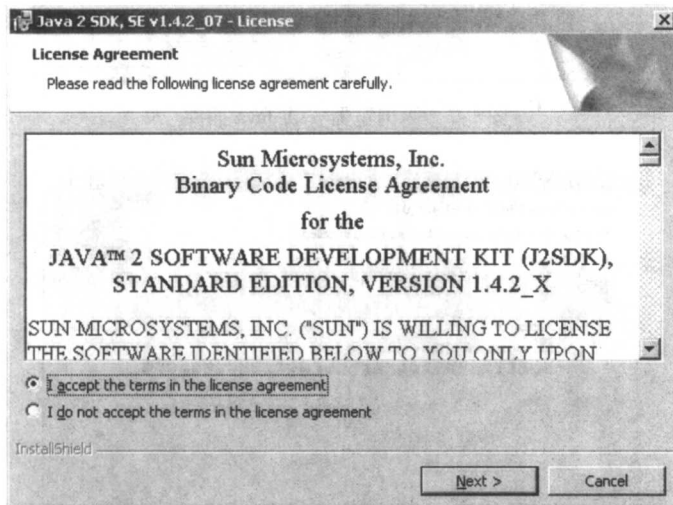


图 1-5 接受 Sun 公司的使用许可条约

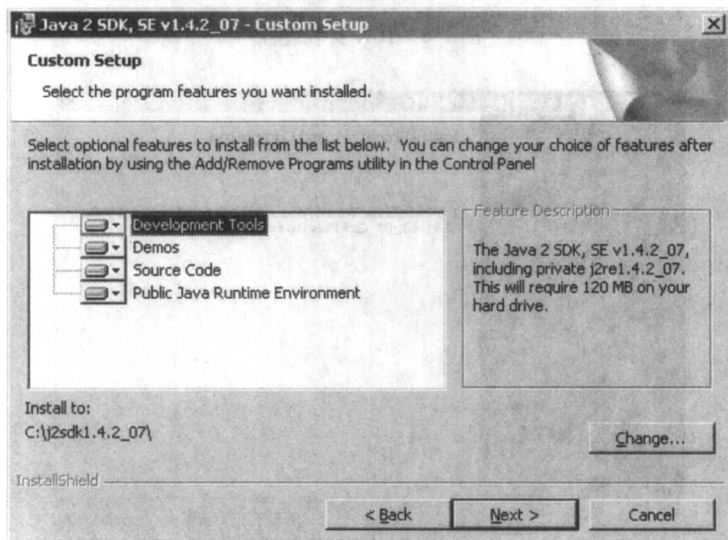


图 1-6 设置 JDK 安装路径以及选择安装组件