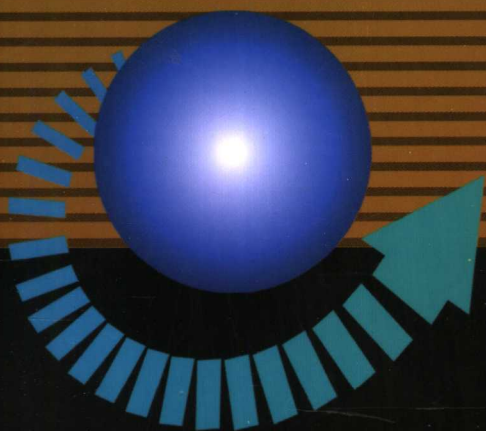




Java

程序设计

陈 强 张 朋 主 编
孙素敏 茅青海 副主编



人民邮电出版社
POSTS & TELECOM PRESS

Java 程序设计

陈 强 张 朋 主 编
孙素敏 茅青海 副主编

人 民 邮 电 出 版 社

图书在版编目 (CIP) 数据

Java 程序设计 / 陈强, 张朋主编. —北京: 人民邮电出版社, 2006.8
ISBN 7-115-15065-6

I. J... II. ①陈...②张... III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 085346 号

内 容 提 要

本书是一本介绍 Java 编程基础知识的教材, 系统地介绍 Java 语言的主要功能及用法。全书共分为 11 章, 内容涵盖: Java 概述、Java 语言基础、Java 面向对象编程、系统基本类库、Java 异常、图形用户界面与 AWT、基于 Swing 的图形用户界面设计、线程、文件和输入输出操作、网络编程和 Applet 程序设计等。本书从 Java 的体系结构、平台标准及主要技术开始讲起, 结合面向对象的编程思想, 详尽地介绍 Java 在各个方面的应用, 并含有大量精心设计的代码实例。书中每章后面给出了大量习题, 帮助读者巩固本章所学知识, 进一步提高 Java 编程水平。

本书主要面向初中级读者, 适合作为高等院校计算机与通信等相关专业的教材, 也可供从事 Java 编程的相关工作人员学习和参考。

编者的话

随着信息技术的高速发展，计算机软件在人们生活中扮演着越来越重要的角色。计算机程序设计技术和方法也不断进步，其间经历了从面向过程到面向对象的重大变革。面向对象的程序设计技术已经成为当今计算机应用开发领域的主流技术。面向对象的程序设计语言也如雨后春笋般不断诞生、发展和壮大。Java 语言自 1995 年诞生以来，至今已历经十年磨砺，成为当前计算机应用较为广泛的面向对象的程序设计语言之一。

本书的特点是熟悉学习对象的同时，力求做到突出重点、详析难点、解答疑点，使读者学习时容易理解和掌握；语言通俗，由浅入深，简明实用，适于自学；通过大量的例题来解释相关的概念和方法，将明确概念和着重实用相结合，有助于读者对基本概念和方法的学习；每章备有大量的习题，可引导读者掌握本章内容并检查学习情况，有助于读者较快地掌握所学的知识。

本书将重点介绍 Java 的基本知识，通过这些基本的语法讲解，使读者了解面向对象的编程思想。全书共分 11 章，各章内容如下。

第 1 章介绍 Java 的起源历史、JDK 的安装配置和 Java 常用的开发工具。

第 2 章介绍 Java 语言基础，包括数据类型、数组、运算符和表达式以及流程控制语句等内容。

第 3 章介绍面向对象的基本概念和特性、Java 中类和接口的概念以及方法的重载和覆盖。

第 4 章介绍 Java 系统基本类库。

第 5 章介绍 Java 异常的相关知识，包括异常概念、异常分类、异常处理机制和自定义异常。

第 6 章介绍 Java AWT 的相关概念、事件处理模型的相关概念与处理方法、常用组件的使用方法。

第 7 章介绍 Java Swing 中常用的类。

第 8 章介绍 Java 线程的相关知识，包括线程概念模型、线程体实现方式、主线程、守护线程以及同步的概念和方法。

第 9 章介绍 Java 语言的文件的概念及流输入/输出（I/O）操作。

第 10 章介绍 Java 的网络编程，主要讲解 Socket 基础知识，包括 Socket 的类型和创建，以及如何创建服务器端和客户端程序。

第 11 章，Applet 编程，介绍如何使用 Applet 运行一个 Java 程序，如何使用 HTML 给 Applet 传递参数。

本书由南昌理工学院的陈强、张朋任主编，孙素敏、茅青海任副主编。其中第 1 章由张朋编写，第 2 章由刘丽君编写，第 3、4、11 章由陈强编写，第 5、6 章由王燕编写，第 7 章由刘福道编写，第 8 章由江静编写，第 9 章由刘雅娟编写，第 10 章由熊丽婷编写。全书由陈强负责进行编排审稿。此外，参与编写的还有谭晓芳、刘肃平、杨波、徐翠莲和陈小梅等。在本书的编写和出版过程中还得到了许多亲友和编辑的帮助，在这里我们要感谢他们。正是

他们的大力支持使本书最终得以出版。

鉴于作者水平有限，加之时间仓促，虽然几经修改但书稿中仍不免会有疏忽和纰漏，请读者提出指正和宝贵意见。

我们联系方式是：cqjxnc@sina.com.cn。

编 者

2006年6月

目 录

第 1 章 Java 概述	1
1.1 Java 语言简介	1
1.1.1 Java 的发展历史	1
1.1.2 Java 的特点	2
1.2 Java 语言的开发平台	4
1.2.1 JDK 与 JDK 的安装	4
1.2.2 JDK 的设置	6
1.2.3 JDK1.5 编译器的新规定	7
1.2.4 Java 的常用开发工具	7
1.3 Java 的程序	9
1.3.1 开发 Java 程序的步骤	9
1.3.2 Java 的第一个程序	9
1.3.3 编译和运行 Java 程序	11
习题	14
第 2 章 Java 语言基础	15
2.1 标识符和关键字	15
2.1.1 标识符	15
2.1.2 关键字	15
2.1.3 分隔符	16
2.2 基本数据类型	16
2.2.1 整型	16
2.2.2 浮点型	17
2.2.3 布尔型	17
2.2.4 字符型	18
2.2.5 常量和变量	18
2.2.6 数据类型转换	20
2.2.7 数据的输入与输出	21
2.3 运算符、表达式和语句	22
2.3.1 运算符和表达式	22
2.3.2 语句	28
2.4 程序流程控制语句	30
2.4.1 程序控制结构	30
2.4.2 分支语句	31

2.4.3 循环语句	35
2.4.4 跳转语句	38
2.4.5 循环语句的嵌套	40
2.5 数组	40
2.5.1 一维数组	40
2.5.2 二维数组	41
2.6 字符串	43
2.6.1 字符串常量	44
2.6.2 字符串变量	44
2.6.3 String 类和 StringBuffer 类的常用方法	45
习题	48
第 3 章 面向对象程序设计	53
3.1 面向对象的基本概念	53
3.2 类和对象	54
3.2.1 类	54
3.2.2 对象	56
3.3 构造方法与方法的重载	58
3.3.1 构造方法	58
3.3.2 方法重载	60
3.4 类的封装、继承和多态	61
3.4.1 封装	61
3.4.2 继承	64
3.4.3 多态	69
3.5 抽象方法和抽象类	71
3.5.1 抽象方法	71
3.5.2 抽象类	71
3.6 最终类和最终方法	73
3.6.1 最终类	73
3.6.2 最终方法	73
3.7 内部类和匿名类	73
3.7.1 内部类	73
3.7.2 匿名类	75
3.8 接口和包	76
3.8.1 接口	76
3.8.2 包	79
习题	81
第 4 章 系统基本类库	89
4.1 Java 类库结构	89
4.2 java.lang 包中的常用类	90

4.2.1	Object 类	90
4.2.2	Math 类	91
4.2.3	数据类型类	92
4.3	java.util 包中的常用类	93
4.3.1	Date 类	93
4.3.2	Calendar 类	94
4.4	Java 帮助文档及其查阅方法	96
	习题	97
第 5 章	异常处理	98
5.1	异常处理机制	98
5.1.1	问题的出现	98
5.1.2	异常处理机制的过程	100
5.2	异常分类	102
5.3	异常处理方法	103
5.3.1	抛出异常	103
5.3.2	捕获及处理异常	106
5.3.3	创建自己的异常类	110
	习题	111
第 6 章	图形用户界面与 AWT	120
6.1	图形用户界面与 AWT	120
6.2	创建图形界面	121
6.2.1	抽象窗口工具集 AWT	121
6.2.2	Java 的图形类结构	121
6.2.3	框架与面板	123
6.3	AWT 常用组件	125
6.3.1	按钮 (Button)	125
6.3.2	标签 (Label)	125
6.3.3	文本框和文本区	126
6.3.4	复选框和单选按钮	128
6.3.5	选择框 (Choice)	130
6.3.6	列表框 (List)	131
6.4	布局管理器类 (Layout)	132
6.4.1	FlowLayout	133
6.4.2	BorderLayout	133
6.4.3	GridLayout	135
6.4.4	CardLayout	135
6.4.5	GridBagLayout 布局管理器	138
6.4.6	null 布局管理器	138
6.4.7	容器的嵌套	140

6.5 事件处理	141
6.5.1 AWT 事件处理模型	141
6.5.2 事件类	143
6.5.3 事件监听器	143
6.5.4 AWT 事件及其相应的监听器接口	144
6.5.5 事件适配器	147
6.5.6 用内部类实现事件处理	147
6.6 菜单组件	151
6.7 对话框	155
6.7.1 对话框 (Dialog)	155
6.7.2 文件对话框 (FileDialog)	157
6.8 图形设计	161
6.8.1 在组件上绘图	161
6.8.2 绘图类 (Graphics)	164
习题	167
第 7 章 基于 Swing 的图形用户界面设计	176
7.1 Swing 简介	176
7.1.1 Swing 概述	176
7.1.2 Swing 的类层次结构	177
7.1.3 Swing 的特性	178
7.1.4 Swing 程序结构简介	180
7.2 Swing 组件和容器	181
7.2.1 Swing 组件的分类	181
7.2.2 使用 Swing 组件的基本规则	182
7.2.3 各种容器面板	182
7.3 JFrame 类和 JPanel 类	183
7.3.1 JFrame 类	184
7.3.2 JPanel 类	184
7.4 Icon 类和 Border 类	185
7.4.1 Icon 类	185
7.4.2 Border 类	186
7.5 标签与按钮组件	187
7.5.1 标签 (JLabel)	187
7.5.2 按钮 (JButton)	187
7.5.3 触发按钮 (JToggleButton)	188
7.5.4 具有图像动态变化的按钮设计	188
7.6 文本输入组件	189
7.6.1 JTextField 类和 JTextArea 类	190
7.6.2 JPasswordField 类	190

7.6.3 JEditorPane 类	190
7.7 复选框与单选按钮组件	191
7.7.1 复选框 (JCheckBox)	191
7.7.2 单选框 (JRadioButton)	191
7.8 列表组件	193
7.8.1 列表 (JList)	193
7.8.2 下拉列表 (JComboBox)	194
7.9 表格和树组件	195
7.9.1 表格 (JTable)	195
7.9.2 树 (JTree)	197
7.10 滑动杆、计时器和进度条组件	200
7.10.1 滑动杆 (JSlider)	200
7.10.2 计时器 (Timer)	201
7.10.3 进度条 (JProgressBar 和 ProgressMonitor)	201
7.11 菜单和工具栏	203
7.11.1 菜单	203
7.11.2 工具栏 (JToolBar)	204
7.12 对话框和文件选择器	207
7.12.1 对话框 (JDialog)	207
7.12.2 文件选择框 (JFileChooser)	208
7.12.3 颜色选择框 (JColorChooser)	209
7.13 Swing 布局管理器	212
7.14 Swing 的事件处理机制	213
习题	222
第 8 章 线程	224
8.1 线程的概述	224
8.2 多线程实现的两种方法	225
8.2.1 继承 Thread 类	225
8.2.2 实现 Runnable 接口	227
8.2.3 两种方法的区别	228
8.2.4 线程组	228
8.3 线程的控制与调度	228
8.3.1 线程的调度与优先级	229
8.3.2 线程生命周期及各状态的相互转换	229
8.3.3 线程的控制	231
8.4 线程的同步机制	235
8.4.1 线程的互斥锁	235
8.4.2 线程同步运行	236
习题	239

第 9 章 文件和输入输出操作	242
9.1 文件	242
9.1.1 文件 File 类	242
9.1.2 随机访问文件 RandomAccessFile 类	248
9.2 数据流	251
9.2.1 数据流的基本概念	251
9.2.2 Java 的标准数据流	251
9.3 字节流初步	252
9.3.1 输入和输出流 InputStream 和 OutputStream	252
9.3.2 文件字节流 FileInputStream 和 FileOutputStream	253
9.4 字节流	255
9.4.1 数据流 DataInputStream 和 DataOutputStream	255
9.4.2 对象流 ObjectInputStream 和 ObjectOutputStream	257
9.4.3 管道流 PipedInputStream 和 PipedOutputStream	257
9.4.4 数组流 ByteArrayInputStream 和 ByteArrayOutputStream	258
9.5 字符流	259
9.5.1 字符流 Reader 和 Writer	259
9.5.2 文件字符流 FileReader 和 FileWriter	260
9.5.3 字符缓冲流 BufferedReader 和 BufferedWriter	261
9.5.4 字符串流 StringReader 和 StringWriter	263
习题	264
第 10 章 网络通信	268
10.1 网络运行机制	268
10.2 URL 通信	269
10.2.1 URL 的基本概念	269
10.2.2 URL 类	270
10.2.3 URLConnection 类	272
10.2.4 显示 URL 资源中的 html 文件	275
10.3 InetAddress 类	277
10.3.1 获取 Internet 上主机的地址	277
10.3.2 获取本地机的地址	278
10.4 Socket 通信	278
10.4.1 Socket 通信机制的基本概念	278
10.4.2 Socket 通信模式	279
10.4.3 实现 Socket 通信	279
习题	284
第 11 章 Applet 程序设计	286
11.1 Applet 概述	286
11.1.1 Applet 的概念	286

11.1.2 Applet 的特点	286
11.2 Applet 类	287
11.2.1 Applet 的创建	287
11.2.2 Applet 的生命周期	288
11.2.3 Applet 的显示与刷新	289
11.3 Applet 程序的编写	292
11.3.1 Applet 编写的步骤	292
11.3.2 超文本标记语言	292
11.3.3 HTML 中嵌入 Applet	293
11.4 Applet 的图形化用户界面 GUI	295
11.4.1 在 Applet 中使用 AWT 组件	295
11.4.2 在 Applet 中使用 Swing 组件	297
11.5 Applet 的多媒体	305
11.5.1 显示图像	305
11.5.2 播放声音	305
11.6 Applet 的通信	310
11.6.1 同页面 Applet 之间的通信	310
11.6.2 Applet 与浏览器之间的通信	314
习题	316

第 1 章

Java 概述

本章主要介绍 Java 语言的发展过程, Java 语言的特点, Java 应用程序和 Applet 小应用程序, Java 工具包 JDK, Java 程序的编辑、编译和运行, 以及 Java 开发工具的介绍和使用等内容。通过本章的学习, 读者可对 Java 语言有个初步的了解和认识。

1.1 Java 语言简介

1.1.1 Java 的发展历史

Java 是一个由 Sun 公司开发而成的新一代编程语言。使用它可在各式各样不同种机器、不同种操作平台的网络环境中开发软件。不论你使用的是哪一种 WWW 浏览器, 哪一种计算机, 哪一种操作系统, 只要 WWW 浏览器上面注明了“支持 Java”, 你就可以看到生动的主页。Java 正在逐步成为 Internet 应用的主要开发语言。它彻底改变了应用软件的开发模式, 带来了自 PC 机以来的又一次技术革命, 为迅速发展的信息世界注入了新的活力。

Sun 的 Java 语言开发小组成立于 1991 年, 其目的是开拓消费类电子产品市场, 为家用消费电子产品开发一个分布式代码系统, 以便用户将 E-mail 发送给电冰箱、电视机、烤箱等家用电器, 对它们进行控制, 并和它们进行信息交换。Sun 公司内部人员把这个项目称为 Green, 在项目研制初始阶段, 项目组成员准备采用 C++ 语言开发该系统, 但是遇到了前所未有的挑战, 因为它太复杂而且安全性差。所以最后项目组成员基于 C++ 开发了一种新的语言 Oak (橡树), 这就是 Java 语言的前身。

Oak 是一种用于网络的精巧而又安全的语言, Sun 公司曾依此投标了一个交互式电视节目, 但结果是被 SGI 打败。正当 Oak 无家可归之时, Mark Ardreesen 开发的 Mosaic 和 Netscape 启发了 Oak 项目组成员, 他们用 Java 编制了 HotJava 浏览器, 并得到了 Sun 公司首席执行官 Scott McNealy 的支持, 开启了 Java 进军 Internet 的契机。

1994 年, 随着 Internet 的崛起、Java 的诞生, 巨型的、臃肿的应用软件开始向小型化发展, 由众多“生活”在 Internet 上的小应用程序 (Applet) 相互协作完成信息的处理与传递。Java 会加速应用软件的小型化、网络化的趋势。随着 Internet 的发展, 软件必然是面向“大众”, 而不是“少数人”的奢侈品, 薄利多销将是网络应用软件的重要特点。分布式对象技术保证了用多少, 下载多少。WWW 的快速增长, 促进了 Java 语言研制的进展, 使得它逐渐成为 Internet 上受欢迎的开发与编程语言。1995 年 5 月 Sun 公司正式发布 Java 的第一个

办公版本。

Java 作为新一代的面向对象的程序设计语言，其平台无关性直接威胁到 Intel 的垄断地位。一些著名的计算机公司纷纷购买了 Java 语言的使用权，如 IBM、Netscape、Novell、Apple、DEC、SGI、Oracle 等公司，甚至包括最不情愿的 Microsoft 公司，都购买了 Java 的使用权。

Java 语言被美国的著名杂志 PC Magazine 评为 1995 年十大优秀科技产品（计算机类就此一项入选）。Microsoft 公司总裁 Bill Gates 不无感慨地说：“Java 是长时间以来最卓越的程序设计语言”。

Java 为 Internet 和 WWW 开辟了一个崭新的时代。万维网（WWW）的创始人 Berners Lee 说：“计算机事业发展的下一个浪潮就是 Java，并且将很快会发生”。甚至有人预言：Java 将是网络上的“世界语”，今后所有的用其他语言编写的软件统统都要用 Java 语言来改写。

注意：Java 的取名也有一个趣闻，有一天，几位 Java 成员组的会员正在讨论给这个新的语言取什么名字，当时他们正在咖啡馆喝着爪哇产的咖啡，Java 是印度尼西亚的一个重要的盛产咖啡的岛屿，中文名叫爪哇，有一个人灵机一动说就叫 Java 怎样，得到了其他人的赞赏，于是，Java 这个名字就这样传开了。开发人员为这种新的语言起名为 Java，其寓意是为世人端上一杯热咖啡。

1.1.2 Java 的特点

Java 是一个广泛使用的网络编程语言，是一种新的计算概念。

首先，作为一种程序设计语言，它简单，面向对象，不依赖于机器的结构，具有可移植性、健壮性、安全性，并且提供了并发的机制，具有很高的性能。

其次，它最大限度地利用了网络，Java 的 Applet 小应用程序可在网络上传输而不受 CPU 和环境的限制。

另外，Java 还提供了丰富的类库，使程序设计者可以很方便地建立自己的系统。

1. 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。

2. 简单性

Java 略去了运算符重载、多重继承等模糊的概念，并且通过实现自动无用信息回收，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，加上标准类库和线程的支持也只有 215KB 左右。

3. 健壮性

Java 在编译和运行程序时，都要对可能出现的问题进行检查，以消除错误的产生。它提供自动无用信息收集来进行内存管理，防止程序员在管理内存时容易产生的错误。通过集成的面向对象的异常（Exception）处理机制，在编译时，Java 揭示出可能出现但未被处理的异常，帮助程序员正确地进行选择，以防止系统的崩溃。

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。

4. 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，用户可以通过 URL 地址在网络上很方便地访问其他对象。

Java 支持 WWW 客户机/服务器计算模式，因此，它支持这两种分布性。对于前者，Java 提供了一个叫做 URL 的对象，利用这个对象，你可以打开并访问具有相同 URL 地址上的对象，访问方式与访问本地文件系统相同。对于后者，Java 的 Applet 小程序可以从服务器下载到客户端，即部分计算在客户端进行，提高系统执行效率。Java 提供了一整套网络类库，开发人员可以利用类库进行网络程序设计，方便地实现 Java 的分布式特性。

5. 安全性

由于 Java 主要用于网络应用程序开发，因此对安全性有较高的要求。如果没有安全保证，用户从网络下载程序执行就非常危险。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，这样可以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。

Java 通过自己的安全机制防止了病毒程序的产生和下载程序对本地系统的威胁破坏。当 Java 字节码进入解释器时，首先，必须经过字节码校验器的检查；然后，Java 解释器将决定程序中类的内存布局；随后，类装载器负责把来自网络的类装载到单独的内存区域，避免应用程序之间相互干扰破坏；最后，客户端用户还可以限制从网络上装载的类只能访问某些文件系统。上述几种机制结合起来，使得 Java 成为安全的编程语言。

6. 高性能

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的信息，使得连接过程更加简单。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，从而具有较高的性能。

7. 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，而不需要采用全局的事件循环机制，这样就很容易地实现网络上的实时交互行为。

8. 平台无关性

Java 的平台无关性是指用 Java 写的应用程序不用修改就可可在不同的软硬件平台上运行。平台无关有两种：源代码级和目标代码级。Java 主要靠 Java 虚拟机 (JVM) 在目标代码级实现平台无关性。JVM 是一种抽象机器，它附在具体操作系统之上，本身具有一套虚机器指令，并有自己的栈、寄存器组等。但 JVM 通常是在软件上而不是在硬件上实现。目前，Sun 系统公司已经设计实现了 Java 芯片，主要使用在网络计算机 NC 上。另外，Java 芯片的出现也会使 Java 更容易嵌入到家用电器中。JVM 是 Java 平台无关的基础，在 JVM 上，有一个 Java 解释器用来解释 Java 编译器编译后的程序。Java 编程人员在编写完软件后，通过 Java 编译器将 Java 源程序编译为 JVM 的字节代码。任何一台机器只要配备了 Java 解释器，就可以运行这个程序，而不管这种字节码是在何种平台上生成的，如图 1.1 所示。另外，Java 采用的是基于 IEEE 标准的数据类型。通过 JVM 保证数据类型的一致性，也确保了 Java 的平台无关性。

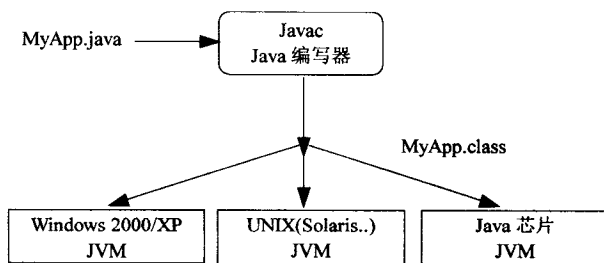


图 1.1 Java 的平台无关性示意图

Java 的平台无关性具有深远意义。首先，它使得编程人员梦寐以求的事情（开发一次软件在任意平台上运行）变成事实，这将大大加快和促进软件产品的开发。其次，Java 的平台无关性正好迎合了“网络计算机”思想。如果大量常用的应用软件（如字处理软件等）都用 Java 重新编写，并且放在某个 Internet 服务器上，那么具有 NC 的用户将不需要占用大量空间安装软件，他们只需要一个 Java 解释器，每当需要使用某种应用软件时，下载该软件的字节代码即可，运行结果也可以发回服务器。目前，已有数家公司开始使用这种新型的计算模式构筑自己的企业信息系统。

9. 灵活性

Java 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。

总之，Java 是一种编程语言、一种开发环境、一种应用环境、一种部署环境、一种广泛使用的网络编程语言，它是一种全新的计算概念。

1.2 Java 语言的开发平台

1.2.1 JDK 与 JDK 的安装

Sun 公司的 Java 不仅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（Java Development Kit, JDK）。开发人员和最终用户都可以利用这个工具集来开发 Java 程序。

Sun 公司在 1996 年早期发布了 Java 第一版。人们很快认识到 Java1.0 并不适合做真正的应用开发。随着 Java1.1 的到来，它很快填补了前者的不足，极大地提高了反射能力并为 GUI 编程增加了新的事件模型。尽管如此，它仍然具有很大的局限性。

1998 年 12 月 Java1.2 版本发布，Sun 公司将其称为 J2SE SDK1.2（Java2 标准版软件开发工具箱 1.2 版）。在 Java 2 平台下，Sun 公司重新定义了 Java 技术的架构，将其分为 3 个版本：

- Java 2 平台 Micro 版（Java 2 Platform Micro Edition, J2ME），适用于小型设备和智能卡；
- Java 2 平台标准版（Java 2 Platform Standard Edition, J2SE），适用于桌面系统；
- Java 2 平台企业版（Java 2 Platform Enterprise Edition, J2EE），适用于以企业为环境

而开发应用程序的专门开发人员。

上述 Java 运行平台都包括了相应的 JVM, JVM 负责将字节码文件(包括程序使用的类库中的字节码)加载到内存,然后采用解释方式来执行字节码文件,即根据相应硬件的机器指令翻译一句执行一句。

目前, Sun 公司已发布了 SDK 的 1.5 版本,读者可在 Sun 公司的 Web 站点免费下载。该站点的 URL 为 <http://java.sun.com/j2se/1.5.0/download.jsp>。下载后的文件为 jdk-1_5_0_01-windows-i586-p.exe,直接运行执行安装。本书的所有程序均在 Sun 公司的 JDK 1.5.0 下调试通过。

JDK 帮助文档(JDK Documentation)在 JDK 的下载页面也有下载链接,建议同时下载 Documentation,它是重要的编程手册,涵盖了对整个 Java 所有方面的内容的描述。

双击 JDK 的安装程序,弹出安装启动界面,如图 1.2 所示。停留一段时间后弹出 JDK 的安装协议窗口,如图 1.3 所示。同意协议后弹出选择 JDK 的安装路径窗口,用户可以自行选择安装路径。然后单击下一步,直到安装完毕为止,如图 1.4、图 1.5、图 1.6 所示。



图 1.2 安装启动界面

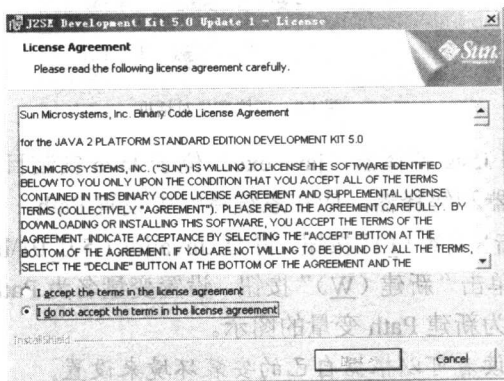


图 1.3 安装协议

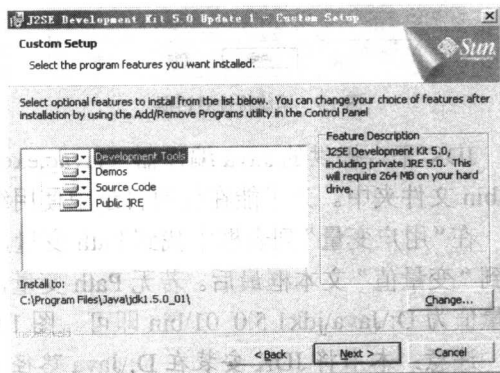


图 1.4 安装目录设置

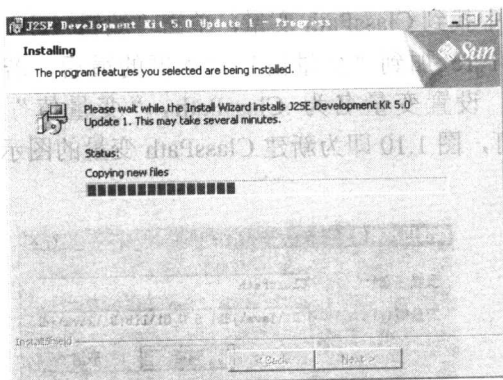


图 1.5 安装进度

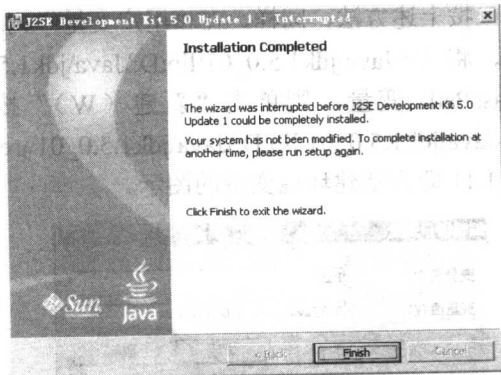


图 1.6 完成 JDK 安装