



附光盘

Java

网络

高级编程

◆ 金勇华 曲俊生 等 编著

快速帮助读者理解Java语言的精髓，掌握利用Java语言的各种编程技巧

突出Java网络程序的开发，包括客户机/服务器、浏览器/服务器结构程序的编写

突出Java小应用程序Applet的开发，使用户的WWW页面更加丰富多彩

详细讲述Java服务器端小应用程序(Servlet)的开发过程

人民邮电出版社
www.pptph.com.cn



00128571

TP312JA



Java 网络 高级编程

◆ 金勇华 曲俊生 等 编著



JSSP/05

人民邮电出版社



北航 C0546838

图书在版编目(CIP)数据

Java 网络高级编程/金勇华,曲俊生编著. —北京:人民邮电出版社,2001.4

ISBN 7-115-09207-9

I .J... II .①金...②曲... III .JAVA 语言—程序设计 IV .TP312

中国版本图书馆 CIP 数据核字(2001)第 18886 号

Java 网络高级编程

- ◆ 编 著 金勇华 曲俊生 等
责任编辑 张立科
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@pptph.com.cn
网址 <http://www.pptph.com.cn>
读者热线:010-67129212 010-67129211(传真)
北京汉魂图文设计有限公司制作
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本:787×1092 1/16
印张:32
字数:774 千字 2001 年 4 月第 1 版
印数:1-5 000 册 2001 年 4 月北京第 1 次印刷

ISBN 7-115-09207-9/TP·2152

定价:54.00 元(含光盘)

内 容 提 要

本书主要向读者讲述了 Java 语言的高级编程技巧。首先介绍了 Java 语言的基本语法，然后又指导已经对 Java 语言具有一定基础的读者进行高级程序设计（网络编程、多线程、JavaBean 组件设计、程序异常处理以及高级数据库程序设计等），并详细讲解了如何编写 Java 小应用程序 Applet 和服务器端小应用程序 Servlet。

本书包含了基于 JDK1.2 和 Borland Jbuilder 的大量源程序代码，这些程序实例可以很有效地帮助读者学习 Java，使读者尽快掌握 Java 语言。

本书适合于希望使用 Java 进行网络应用程序开发的人员阅读。

前 言

Java 语言在诞生的短短几年时间内，在计算机领域尤其是网络编程方面得到了广泛的应用，并且日益发挥着重要的作用。Java 语言是一种简单、面向对象、分布式、与平台无关、高效能、多线程、动态的语言。这些特点使用户可以利用 Java 语言方便有效地开发出功能强大的网络应用程序。

在本书中我们将重点讲述以下内容：

1. 阐述 Java 语言的精髓，介绍使用 Java 语言进行开发的各种编程技巧；
2. 说明 Java 的优点，例如与平台无关、多线程、异常处理等，以及利用 Java 的这些优点来编写高效的程序；
3. 列举大量的程序实例以帮助读者快速编写 Java 程序；
4. 突出 Java 的网络程序开发，包括客户机/服务器、浏览器/服务器结构的应用程序的编写；
5. 突出 Java 小应用程序 Applet 的开发，使用户的 WWW 页面更加丰富多彩；
6. Java 服务器端小应用程序 Servlet 可以集成到能实现 Java 服务器 API 的任何服务器上，它是实现智能代理的第一步，在本书中详细说明 Servlet 的开发过程；
7. 提供大量的源程序代码，这些程序都尽可能地从业者的角度着想，使用户可以直接根据这些源代码来调试有关程序。

在本书中，我们的目标是让每一章都只讲述一种语言特性或者只讲述少数几个相互关联的特性，从而使读者在从一个主题转向下一个主题时，可以更加容易消化前面学到的知识。

本书共分为下面的 12 章。

第 1 章：Java 概述

这一章主要介绍了 Java 语言的起源和特点，并介绍了如何在各种不同的操作系统环境下配置 Java 的执行环境，以及利用 JDK 或 Borland Jbuilder 来进行 Java 程序的开发和执行工作。

第 2 章：Java 语言的基本概念

这部分介绍了 Java 语言所支持的数据类型和针对数据类型的各种运算，这是 Java 作为程序设计语言的基础部分。其中主要讲述了 Java 语言中的标识符、关键字、常量、变量、数据类型以及运算符等方面的内容。

第 3 章：流程控制

本章介绍了 Java 语言的流程控制。对于 C/C++ 程序员来说，Java 语言并不显陌生，Java 和 C/C++ 的流程控制几乎完全一致（利用 if-else 语句来实现分支选择；利用 for 或 while 语句来实现循环控制等），只不过少了 goto 语句。通过本章的学习，读者就可以开始编写一些简单的 Java 程序。

第 4 章：Java 小应用程序 Applet

Applet 是 Java 语言最重要的应用之一，利用 Applet 可以大大丰富 WWW 的内容，扩展网页的功能。在这一章中，我们详细地说明 Java Applet 的基本结构（init、start、stop 和 destroy

方法)、高级 Applet 代码、Applet 编程技巧以及利用浏览器来浏览 Applet 的方法。

第 5 章：服务器端小应用程序 Servlet

Servlet 是随着 Java 技术逐渐发展起来的，它与 Applet 不同，是完全在服务器端解释执行的，因此它可以弥补 Applet 的很多不足。本章说明了如何配置 Servlet 运行环境，并主要讲述了 Servlet 的编程技巧。

第 6 章：Java 的多线程机制

本章主要介绍了多线程的基本概念，探讨了 Java 语言用以实现多线程的两种机制；对 Thread 和 ThreadGroup 类进行了详细叙述，并着重说明了线程的 4 个生命周期：新建、可运行、死和堵塞。最后还说明了线程竞争的概念。

第 7 章：流和文件

本章主要讲述了流操作和文件操作的基本知识。流操作主要介绍了基本输入输出流中的 InputStream 类、OutputStream 类、PrintStream 类以及 System.in 和 System.out 对象。文件操作主要讲述了 FileInputStream 类、FileOutputStream 类、File 类和 RandomAccessFile 类。

第 8 章：Java 网络程序设计

在 Java 语言编程中，网络编程是十分重要的。利用强大的网络类库来编写客户机/服务器 (C/S) 程序以及扩充 Java 小应用程序 Applet 来编写浏览器/服务器 (B/S) 程序是 Java 语言的主要应用。本章讲述了网络类包 java.net 中的网络类如 URL、URLConnection、HttpURLConnection、Socket、ServerSocket、InetAddress 等类的基本用法；详细说明了如何编写客户机/服务器程序和浏览器/服务器程序，并在最后说明了如何利用回调技术来编写客户机/服务器程序，从而使用户可以在一台机器上实现网络程序的编写。

第 9 章：异常处理

在程序设计和运行过程中，异常和出错的类型是多种多样的。在程序设计过程中，异常是不可避免的，而错误可以在应用程序发布之前被发现。对于不同的异常和错误类型必须进行适当的处理，这样程序才能顺利地运行。能否良好地处理异常和错误已经成为一个应用程序开发好坏标准之一。本章主要讲述常见的出错和异常的各种情况，以及出现异常后利用 try、catch、finally 等关键字对异常进行处理的方法。并在最后列出了异常的基本类型。

第 10 章：JavaBean 组件设计

本章主要向读者介绍了 JavaBean 的基本技术。首先介绍了 JavaBean 的基本概念，包括 JavaBean 的基本特性、JavaBean 的方法、属性和事件。然后说明了 JavaBean 中的 4 种属性，包括单值属性、索引属性、关联属性和限制属性，并详细地说明它们之间的区别和实现，其中主要讲述了如何实现关联属性和限制属性的属性通知和监听机制。最后介绍了 JavaBean 中的事件发送和监听机制。

第 11 章：JDBC 数据库

数据库的支持对于程序设计语言的重要性是不言而喻的。在 Java 语言中，利用 JDBC 可以方便地建立同数据库的连接以及对数据库进行各种所需的操作。JDBC 与 ODBC 基本相同，作为连接数据库的数据连接使得 Java 程序可以支持多种不同的数据库。

本章中主要讲述了如何利用 JDBC 来实现对数据库的访问，建立同指定数据库的连接，同时利用 JDBC 语句来执行相应的 SQL 语句，从而实现对数据库的操作。最后给出了一个对数据库操作的实例来帮助读者更好地理解数据库程序的设计。

第 12 章：高级编程实例

本章通过 9 个比较有代表意义的实例来详细说明 Java 语言的高级编程技术，通过实例可以使读者更好地理解本书的内容，快速地领会 Java 语言的精髓。

本书的读者对象是从事 Java 语言编程的专业人员。由于时间比较紧促，加之作者水平有限，书中错误在所难免，望读者给予批评指正，多提宝贵意见。

作者
2001/3/16

目 录

| | |
|----------------------------------|----|
| 第 1 章 Java 概述 | 1 |
| 1.1 Java 介绍 | 1 |
| 1.1.1 Java 的起源 | 1 |
| 1.1.2 Java 的特点 | 1 |
| 1.1.3 Java 和 C++ 的区别 | 3 |
| 1.2 Java 的开发和执行环境 | 3 |
| 1.2.1 JDK 的下载 | 4 |
| 1.2.2 JDK 的安装与环境配置 | 4 |
| 1.2.3 JDK 的环境工具及其使用 | 5 |
| 1.2.4 JDK 的使用 | 9 |
| 1.2.5 Borland Jbuilder 的使用 | 11 |
| 1.3 本章小结 | 19 |
| 第 2 章 Java 语言的基本概念 | 20 |
| 2.1 Java 语言基础 | 20 |
| 2.2 Java 标识符与关键字 | 21 |
| 2.2.1 标识符 | 22 |
| 2.2.2 关键字 | 22 |
| 2.3 常量 (Literals) | 23 |
| 2.3.1 整数 | 23 |
| 2.3.2 浮点数 | 23 |
| 2.3.3 布尔型常量 | 23 |
| 2.3.4 字符型常量 | 24 |
| 2.3.5 字符串常量 | 24 |
| 2.4 变量声明 | 25 |
| 2.4.1 声明整数类型 | 25 |
| 2.4.2 声明浮点类型 | 25 |
| 2.4.3 声明明字符类型 | 26 |
| 2.4.4 声明布尔类型 | 26 |
| 2.4.5 声明数组 | 26 |
| 2.5 变量的作用域 | 26 |
| 2.6 运算符及其优先级 | 27 |
| 2.6.1 整数运算符 | 28 |
| 2.6.2 布尔运算符 | 31 |
| 2.6.3 浮点数运算符 | 33 |

| | | |
|--------------|------------------------------------|-----------|
| 2.6.4 | 数组运算符 | 33 |
| 2.6.5 | 字符串运算符 | 34 |
| 2.7 | 类型转换 | 35 |
| 2.8 | 注释 | 35 |
| 2.9 | 本章小结 | 37 |
| 第 3 章 | 流程控制 | 38 |
| 3.1 | 概述 | 38 |
| 3.2 | 条件语句 | 38 |
| 3.2.1 | 简单条件语句 | 38 |
| 3.2.2 | 复合条件语句 | 39 |
| 3.3 | 循环语句 | 40 |
| 3.3.1 | while 语句 | 40 |
| 3.3.2 | do 语句 | 41 |
| 3.3.3 | for 语句 | 41 |
| 3.4 | 开关语句 | 42 |
| 3.5 | 转移语句 | 45 |
| 3.6 | 返回语句 | 47 |
| 3.7 | 本章小结 | 48 |
| 第 4 章 | Java 小应用程序 Applet | 49 |
| 4.1 | 什么是 Applet | 49 |
| 4.1.1 | Applet 基础 | 49 |
| 4.1.2 | Applet 的安全基础 | 51 |
| 4.1.3 | Applet 示例 | 51 |
| 4.2 | Applet 类 | 55 |
| 4.3 | Applet 的结构 | 56 |
| 4.3.1 | init() 方法 | 56 |
| 4.3.2 | start() 方法 | 56 |
| 4.3.3 | stop() 方法 | 57 |
| 4.3.4 | destroy() 方法 | 57 |
| 4.4 | 高级 Applet 代码 | 57 |
| 4.4.1 | 利用 Applet 来接收从 HTML 中传递过来的参数 | 57 |
| 4.4.2 | 利用 Applet 来显示图像 | 62 |
| 4.4.3 | 利用 Applet 引入声音 | 66 |
| 4.4.4 | Applet 之间进行通信 | 70 |
| 4.5 | Applet 编程技巧 | 77 |
| 4.5.1 | Applet 中的鼠标操作 | 77 |
| 4.5.2 | Applet 中的键盘操作 | 81 |
| 4.6 | 利用浏览器浏览 Applet | 86 |

| | | |
|--------------|-------------------------------------|-----------|
| 4.6.1 | 浏览 Applet | 86 |
| 4.6.2 | 查看 Java 记录 | 88 |
| 4.7 | 本章小结 | 89 |
| 第 5 章 | 服务器端小应用程序 Servlet | 90 |
| 5.1 | 传统的计算模式 | 90 |
| 5.1.1 | 集中式计算模式 | 90 |
| 5.1.2 | 胖客户机/服务器计算模式 | 90 |
| 5.1.3 | 瘦客户机/服务器计算模式 | 92 |
| 5.1.4 | Java Desktop 与 X 终端 | 92 |
| 5.2 | Java 计算 | 92 |
| 5.3 | Java Servlet 的结构 | 94 |
| 5.4 | Servlet 及其功能 | 95 |
| 5.4.1 | 什么是 Servlet | 95 |
| 5.4.2 | Servlet 与 Applet 的区别 | 96 |
| 5.4.3 | Servlet 的应用 | 96 |
| 5.5 | Servlet 与 CGI 的比较 | 96 |
| 5.5.1 | CGI | 96 |
| 5.5.2 | Servlet 的优点 | 97 |
| 5.5.3 | Servlet 与 CGI-BIN | 98 |
| 5.5.4 | Servlet 与 CGI 程序的比较 | 98 |
| 5.6 | 运行 Servlet | 99 |
| 5.7 | 使用 Java Server 与 Servlet 开发工具 | 101 |
| 5.7.1 | Java Web Server 的安装与启动 | 101 |
| 5.7.2 | Servlet 的加载和调用 | 104 |
| 5.7.3 | JSDK 的安装与使用 | 105 |
| 5.8 | GenericServlet 编程 | 109 |
| 5.8.1 | Servlet 编程的一个小例子 | 109 |
| 5.8.2 | Servlet 中的对象 | 112 |
| 5.8.3 | Servlet 输出 HTML 文件 | 117 |
| 5.9 | HttpServlet 编程介绍 | 118 |
| 5.9.1 | HTTP 简介 | 118 |
| 5.9.2 | HttpServlet 编程入门 | 121 |
| 5.9.3 | Cookie 编程介绍 | 126 |
| 5.10 | Servlet 高级编程 | 133 |
| 5.10.1 | Servlet 与 Form | 133 |
| 5.10.2 | 编写一个 SHTML 文本 | 144 |
| 5.10.3 | 异常处理 | 146 |
| 5.10.4 | Servlet 与 Applet 的通信 | 146 |
| 5.11 | 一个综合的例子 | 152 |

| | | |
|--------------|---------------------------------|------------|
| 5.11.1 | 聊天室客户端程序 | 152 |
| 5.11.2 | “聊天室”服务器程序 | 163 |
| 5.11.3 | “聊天” | 174 |
| 5.12 | 本章小结 | 175 |
| 第 6 章 | Java 的多线程机制 | 176 |
| 6.1 | 什么是多线程机制 | 176 |
| 6.2 | 多线程实现机制 | 176 |
| 6.2.1 | 继承 Thread 类 | 176 |
| 6.2.2 | 实现 Runnable 界面 | 177 |
| 6.3 | Thread 类 | 178 |
| 6.3.1 | yield 方法 | 179 |
| 6.3.2 | 守护线程 | 180 |
| 6.3.3 | 变量 | 180 |
| 6.4 | ThreadGroup 类 | 180 |
| 6.5 | 线程的生命周期 | 193 |
| 6.5.1 | 新建 (new) | 193 |
| 6.5.2 | 可运行 (runnable) | 193 |
| 6.5.3 | 死 (dead) | 193 |
| 6.5.4 | 堵塞 (blocked) | 193 |
| 6.6 | 线程竞争 | 200 |
| 6.7 | 本章小结 | 204 |
| 第 7 章 | 流和文件 | 205 |
| 7.1 | 什么是流 | 205 |
| 7.2 | 基本输入输出流 | 206 |
| 7.2.1 | InputStream 类 | 206 |
| 7.2.2 | OutputStream 类 | 207 |
| 7.2.3 | System.in 和 System.out 对象 | 207 |
| 7.2.4 | PrintStream 类 | 208 |
| 7.3 | 文件操作类 | 209 |
| 7.3.1 | FileInputStream 类 | 209 |
| 7.3.2 | FileOutputStream 类 | 210 |
| 7.3.3 | File 类 | 211 |
| 7.3.4 | RandomAccessFile 类 | 213 |
| 7.4 | 读写文件实例 | 215 |
| 7.4.1 | 创建项目并设计主界面 | 215 |
| 7.4.2 | 实现浏览文件得到源和目标文件或文件夹 | 220 |
| 7.4.3 | 准备拷贝 | 224 |
| 7.4.4 | 完成 PrepareCopyFile 线程类 | 229 |

| | | |
|---------------|---|------------|
| 7.4.5 | 完成 CopyFile 类 | 232 |
| 7.5 | 本章小结 | 233 |
| 第 8 章 | Java 网络程序设计 | 234 |
| 8.1 | 网络基本知识概述 | 234 |
| 8.2 | java.net 包 | 234 |
| 8.2.1 | URL 类 | 235 |
| 8.2.2 | URLConnection 类 | 236 |
| 8.2.3 | HttpURLConnection 类 | 237 |
| 8.2.4 | Socket 类 | 238 |
| 8.2.5 | 设置 Socket 的选项 | 239 |
| 8.2.6 | ServerSocket 类 | 241 |
| 8.2.7 | InetAddress 类 | 245 |
| 8.2.8 | DatagramSocket 和 DatagramPacket 类 | 245 |
| 8.3 | 编写客户机/服务器程序 | 250 |
| 8.3.1 | 客户机/服务器工作模式 | 250 |
| 8.3.2 | 客户机/服务器编程过程 | 251 |
| 8.4 | 编写浏览器/服务器程序 | 264 |
| 8.5 | 回调技术编写客户机/服务器程序 | 267 |
| 8.6 | 本章小结 | 271 |
| 第 9 章 | 异常处理 | 272 |
| 9.1 | 出错和异常 | 272 |
| 9.2 | try 和 catch | 272 |
| 9.3 | throw | 274 |
| 9.3.1 | 直接抛出异常 | 274 |
| 9.3.2 | 间接抛出异常 | 275 |
| 9.4 | finally | 276 |
| 9.5 | 常见的异常类型 | 277 |
| 9.6 | 本章小结 | 278 |
| 第 10 章 | JavaBean 组件设计 | 279 |
| 10.1 | JavaBean 概述 | 279 |
| 10.1.1 | 方法 | 279 |
| 10.1.2 | 属性 | 279 |
| 10.1.3 | 事件 | 280 |
| 10.2 | JavaBean 属性 | 280 |
| 10.2.1 | 单值属性 | 280 |
| 10.2.2 | 索引属性 | 281 |
| 10.2.3 | 关联属性 | 283 |

| | | |
|---------------|------------------------------|------------|
| 10.2.4 | 限制属性 | 285 |
| 10.2.5 | 属性监听实现 | 288 |
| 10.3 | JavaBean 事件 | 289 |
| 10.4 | JavaBean 示例 | 292 |
| 10.4.1 | 创建一个工程并实例化 JavaBean 容器 | 292 |
| 10.4.2 | 编写 LabelBean | 295 |
| 10.4.3 | 实现 JavaBean 的关联属性支持 | 298 |
| 10.4.4 | 实现关联属性监听 | 301 |
| 10.4.5 | 注册属性变化和事件发生源 | 304 |
| 10.4.6 | 实现 PropertyChange 方法 | 307 |
| 10.4.7 | 实现主界面的按钮事件 | 307 |
| 10.4.8 | 实现 JavaBean 事件发送 | 307 |
| 10.4.9 | 实现事件监听 | 310 |
| 10.4.10 | 实现键盘监听事件接口 | 315 |
| 10.4.11 | 程序运行结果和流程图 | 315 |
| 10.5 | 本章小结 | 324 |
| 第 11 章 | JDBC 数据库 | 325 |
| 11.1 | JDBC 简介 | 325 |
| 11.2 | 数据库类实现 | 325 |
| 11.3 | 访问数据库的 JDBC 类 | 327 |
| 11.3.1 | Connection 类 | 327 |
| 11.3.2 | DatabaseMetaData 类 | 328 |
| 11.3.3 | ResultSetMetaData | 333 |
| 11.4 | JDBC 语句 | 334 |
| 11.4.1 | Statement | 334 |
| 11.4.2 | PreparedStatement | 335 |
| 11.4.3 | CallableStatement | 336 |
| 11.4.4 | ResultSet 类 | 337 |
| 11.5 | 访问数据库实例 | 340 |
| 11.5.1 | 设计主窗口 | 340 |
| 11.5.2 | 连接数据库 | 344 |
| 11.5.3 | 显示数据库数据 | 345 |
| 11.5.4 | 实现增加和修改对话框 | 348 |
| 11.5.5 | 完成主窗口事件 | 350 |
| 11.5.6 | 运行结果和源程序 | 351 |
| 11.6 | 本章小结 | 363 |
| 第 12 章 | 高级编程实例 | 364 |
| 12.1 | 浏览器/服务器方式聊天程序 | 364 |

| | | |
|--------|----------------------------|-----|
| 12.1.1 | 监听新用户加入 | 364 |
| 12.1.2 | 编写用户实例类 | 369 |
| 12.1.3 | 编写监听用户信息类 | 370 |
| 12.1.4 | 控制在线用户 | 372 |
| 12.1.5 | 编写客户端 Applet 类 | 373 |
| 12.1.6 | 编写客户端用户监听线程 | 375 |
| 12.1.7 | 通信流程和运行结果 | 376 |
| 12.1.8 | 源程序清单 | 378 |
| 12.2 | Java 打印 | 394 |
| 12.2.1 | 绘图的实现 | 394 |
| 12.2.2 | 打印的实现 | 396 |
| 12.2.3 | 运行结果和源程序清单 | 397 |
| 12.3 | Java 本地文件——JNI | 405 |
| 12.3.1 | 编写 Java 程序 | 406 |
| 12.3.2 | 将 Java 程序编译成类文件 | 406 |
| 12.3.3 | 产生 JNI 风格的头文件 | 407 |
| 12.3.4 | 实现本地方法 | 407 |
| 12.3.5 | 编译生成动态连接库文件 | 408 |
| 12.3.6 | 运行 Java 程序 | 408 |
| 12.3.7 | 在本地方法中处理异常 | 411 |
| 12.4 | Java Applet 的安全性 | 412 |
| 12.5 | 文件上下载 | 417 |
| 12.5.1 | 设计主窗口 | 417 |
| 12.5.2 | 主窗体事件的实现 | 425 |
| 12.5.3 | 实现连接服务器类 ConnListen | 434 |
| 12.5.4 | 实现下载线程类 DownloadFile | 434 |
| 12.5.5 | 实现上载类 UploadFile | 435 |
| 12.5.6 | 程序运行结果和源程序清单 | 435 |
| 12.6 | 直接发送电子邮件 | 455 |
| 12.6.1 | 在 Applet 中调用工具发送电子邮件 | 456 |
| 12.6.2 | 直接发送电子邮件 | 458 |
| 12.7 | 滚动条幅 Applet | 464 |
| 12.7.1 | 确定控制参数 | 464 |
| 12.7.2 | 编写 Applet 类 | 465 |
| 12.7.3 | 实现滚动线程 | 471 |
| 12.7.4 | 实现 Label 鼠标事件 | 474 |
| 12.7.5 | 源程序清单 | 481 |
| 12.8 | 远程方法调用 | 489 |
| 12.8.1 | 定义一个扩展了 Remote 接口的接口 | 489 |
| 12.8.2 | 定义一个实现上面接口的类 | 489 |

| | | |
|--------|--|-----|
| 12.8.3 | 使用 <code>rmic</code> 程序生成远程实现所需要的存根和结构框架 | 491 |
| 12.8.4 | 创建一个可以对服务器进行 <code>RMI</code> 调用的客户程序 | 492 |
| 12.8.5 | 登记并使用远程服务器和客户程序 | 492 |
| 12.9 | 本章小结 | 493 |

第 1 章 Java 概述

Java 语言的出现使得目前种类繁多的编程语言又增加了一种,然而,这种增加是否有必要?为什么要选择用 Java 语言来进行编程?Java 语言具有哪些优点和特性?Java 语言和 C++ 语言有什么区别?本章中将详细说明上述问题。

1.1 Java 介绍

Java 作为一种面向对象的程序设计语言,从 1995 年 7 月开始在计算机界受到高度重视。Java 降低了 Internet 应用程序的编写难度。在 WWW 世界中有数不清的 Java 小应用程序,它们能够产生一些有趣的效果,使 Internet 世界变得更加丰富多彩。

1.1.1 Java 的起源

1991 年 4 月,由美国 Sun 计算机公司的 Patrick-Naughton, James Gosling 以及 Mike Sheridan 等组成的开发小组开始着手于“Green”工程的实施。这个工程最初的目标是发展消费类电子产品,计划使用 C、C++ 来制作一个软件,以实现家用电器进行集成控制的小型控制装置。后因语言本身和市场的问题,这个技术上非常成功的消费类电子产品在商业上的发展几经失败,无法达到当初预期的目标。

1994 年,WWW 以极快的速度风靡了整个 Internet。“Green”工程的开发小组发现,他们为解决编程困难而设计的新型编程语言 Oak 比较适合于 Internet 程序的编写。于是,他们结合 WWW 的需要,对 Oak 进行了改进和完善,设计出了一种非常适合于编写 Internet 程序的语言。

在为这种语言选择名称时,开发小组经过了无数次的激烈讨论,他们要求语言的名称能够表达出语言本身的动画、速度、交互性方面的特色。“Java”在无数的建议中脱颖而出。出乎意料的是,“Java”不是由几个单词的首字母所组成的,而是从许多程序设计师都钟爱的热腾腾的香浓咖啡中得到的灵感。

就这样,Java 语言诞生了。

1.1.2 Java 的特点

Java 是一种编程语言。因为 Java 能够比较好地适应 Internet 编程,所以有些人认为它和 HTML 和 VRML 等语言一样是一种描述性语言,这种说法是不正确的。Java 作为编程语言所描述的是产生结果的过程,而 HTML 等描述性语言是用于标明内容和位置的。Java 编制的小程序(Applet)在网页中执行后所产生的效果不是简单的描述性语言能够产生的。

在 Sun 公司的“Java 白皮书”中对 Java 有如下的定义：

Java: A simple, Object-oriented, distributed, robust, secure, architecture-neutral, portable, high-performance, multi-threaded, and dynamic language. (Java: 一种简单的、面向对象的、分布式的、健壮的、安全的、结构中立的、可移植的、高效能的、多线程的和动态的语言)。

这个定义充分地表现了 Java 语言的特点。

1. 简单性

Java 是一种简单的语言。Java 取消了其他的许多编程语言中十分繁琐的和难以理解的内容，例如 C/C++ 中的指针、C++ 中类的多重继承等。Java 语言在外观上让大多数程序员都感到很熟悉，这样程序员不需要经过长时间训练就能够学会使用这种新型的程序开发工具。Java 要求的基本解释器约为 40KB，若加上基本的程序库，约为 215KB。由于 Java 程序很小，因此在小型机上 Java 程序也能够很好地执行。

2. 面向对象

Java 是一种面向对象的语言。这里的“对象”是指应用程序的数据及其操作方法。Java 程序的设计思路不同于 C 语言基于过程的程序设计思路。设计面向对象的程序需要将编程重点放在资料和接口上。

3. 分布式特性

Java 是一种分布式的语言。Java 有一个很周全的程序库，且很容易与 HTTP 和 TCP/IP 通信协议相配合。Java 应用程序能在网络上开启并连结使用对象，就如同透过 URL 链接使用一个本地文件系统。利用 Java 来开发分布式的网络程序是 Java 的一个重要应用。

4. 健壮性

用 Java 编写的程序在多种情况下都能稳定地执行。Java 与 C/C++ 最大的不同点是在于 Java 有一个专门的指针模型，它的作用是排除内存中的数据被覆盖和毁损的可能性。

5. 安全性

Java 是一种安全的网络编程语言。Java 是被设计用于网络及分布式的环境中的，所以 Java 程序的安全性自然是十分重要的。Java 拥有数个阶层的互锁保护措施，能有效地防止病毒的侵入和破坏行为的发生。

6. 结构中立性

众所周知，网络是由很多不同机型的计算机组合而成的。这些计算机的 CPU 和操作系统体系结构均有所不同，因此，要使一个应用程序可以在每一种计算机上都能够执行是很难的。所幸的是，Java 的编译器能够产生一种结构中立的文件目标格式，这使得编译码能够在多数的处理器中执行。

7. 可移植性

Java 的源代码和 C/C++ 的源代码一样具有可移植性，而且 Java 的可移植性在相比之下还要更强一些。只要有对应系统的解释器，Java 程序就可以在所有的系统上执行。Java 的原始资料形式的大小是被指定的，例如 Java 之所以一直用“float”来表示一个 32 位字节的 IEEE754 浮点运算数字，正是因为绝大多数的 CPU 都有此共同特征。Java 自带的程序库定义了一些可移植的程序接口，这使 Java 语言具备了很好的可移植性。

8. 解释型语言

Java 的源程序首先由编译器编译成字节码，之后再由解释器解释执行。Java 解释器能直接在任何机器上执行 Java 字节码 (Byte Codes)，因此在进行程序链接时节省了时间，这对于