

北京大学信息技术系列教材

# Java 程序设计

主编 蔡翠平

编著 唐大仕

清华大学出版社

北方交通大学出版社

· 北 京 ·

## 内 容 简 介

本书详细介绍了 Java 程序设计的基本环境、概念和方法。内容分为三个部分：第一部分介绍了 Java 语言基础，包括数据、控制结构、数组、类、包、对象、接口等；第二部分介绍了 Java 深入知识，包括传值调用、虚方法调用、异常处理、工具类与算法；第三部分是 Java 的应用，包括线程、流式文件、AWT 及 Swing 图形用户界面，以及 Java 在网络、多媒体、数据库等方面的应用。

本书内容详尽，循序渐进，在介绍编程技术的同时，还着重讲解了有关面向对象程序设计的基本概念和方法。书中提供了丰富的典型实例，具有可操作性，便于读者的学习与推广应用。各章附有大量习题，便于读者思考和复习。

本书内容和组织方式立足于高校教学教材的要求，同时可作为计算机技术的培训教材，还可作为 Sun 认证考试（SCJP）的考试用书。

**版权所有，翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。**

### 图书在版编目（CIP）数据

Java 程序设计/唐大仕编著. —北京：北方交通大学出版社，2003.3

（北京大学信息技术系列教材）

ISBN 7-81082-099-0

I .J… II.唐… 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（2003）第 005216 号

责任编辑：谭文芳

印刷者：北京东光印刷厂

出版发行：北方交通大学出版社 邮编：100044 电话：010-51686045，62237564

清华大学出版社 邮编：100084

经 销：各地新华书店

开 本：787×1092 1/16 印张：23 字数：556 千字

版 次：2003 年 4 月第 1 版 2003 年 4 月第 1 次印刷

印 数：6000 册 定价：29.00 元

# 北京大学信息技术系列教材

## 编委会成员

主 任：蔡翠平

副主任：吕凤翥

委 员：(以姓氏笔画为序)

尹会滨 许 彦 吕凤翥 任吉治 张亦工

吴筱萌 尚俊杰 林洁梅 周宏滔 陈 虎

赵 文 赵丹群 徐尔贵 唐大仕 蔡翠平

缪 蓉 潘 曦

# 目 录

|   |      |
|---|------|
| <b>第 1 章 Java 语言与面向对象的程序设计</b> .....      | (1)  |
| 1.1 Java 语言简介 .....                       | (1)  |
| 1.1.1 Java 语言出现的背景、影响及应用前景 .....          | (1)  |
| 1.1.2 Java 的特点 .....                      | (2)  |
| 1.1.3 Java 和 C、C++ .....                  | (4)  |
| 1.2 面向对象程序设计.....                         | (5)  |
| 1.2.1 面向对象概述.....                         | (5)  |
| 1.2.2 对象、类与实体 .....                       | (6)  |
| 1.2.3 对象的状态与行为 .....                      | (7)  |
| 1.2.4 对象的关系 .....                         | (7)  |
| 1.2.5 面向对象的软件开发过程 .....                   | (8)  |
| 习题 .....                                  | (9)  |
| <br>                                      |      |
| <b>第 2 章 简单的 Java 程序</b> .....            | (11) |
| 2.1 Application 与 Applet 程序 .....         | (11) |
| 2.1.1 Application 程序 .....                | (11) |
| 2.1.2 Applet 程序 .....                     | (12) |
| 2.1.3 Java 程序的基本构成 .....                  | (13) |
| 2.2 程序的编辑、编译与运行 .....                     | (15) |
| 2.2.1 Java 工具包 JDK .....                  | (15) |
| 2.2.2 Application 的编辑、编译与运行 .....         | (15) |
| 2.2.3 Applet 的编辑、编译与运行 .....              | (19) |
| 2.2.4 使用 jar 打包程序 .....                   | (21) |
| 2.3 Java 程序中的基本输入与输出 .....                | (21) |
| 2.3.1 字符界面的输入与输出 .....                    | (22) |
| 2.3.2 Applet 图形界面输入与输出 .....              | (24) |
| 2.3.3 Java Application 图形界面输入与输出 .....    | (26) |
| 2.3.4 同时作为 Application 与 Applet 的程序 ..... | (27) |
| 2.4 Java 集成开发环境 .....                     | (28) |
| 2.4.1 几种辅助工具的使用 .....                     | (29) |
| 2.4.2 几种集成工具的使用 .....                     | (31) |
| 习题 .....                                  | (34) |

|                                 |      |
|---------------------------------|------|
| <b>第 3 章 数据运算、流控制和数组</b> .....  | (36) |
| 3.1 数据类型、变量与常量.....             | (36) |
| 3.1.1 数据类型.....                 | (36) |
| 3.1.2 标识符.....                  | (37) |
| 3.1.3 常量.....                   | (37) |
| 3.1.4 变量.....                   | (38) |
| 3.1.5 程序的书写与注释.....             | (39) |
| 3.2 运算符与表达式.....                | (41) |
| 3.2.1 算术运算符.....                | (41) |
| 3.2.2 关系运算符.....                | (43) |
| 3.2.3 逻辑运算符.....                | (43) |
| 3.2.4 位运算符.....                 | (44) |
| 3.2.5 赋值与强制类型转换.....            | (48) |
| 3.2.6 条件运算符.....                | (49) |
| 3.2.7 表达式及运算的优先级、结合性.....       | (50) |
| 3.3 流程控制语句.....                 | (50) |
| 3.3.1 结构化程序设计的三种基本流程.....       | (51) |
| 3.3.2 简单语句.....                 | (51) |
| 3.3.3 分支语句.....                 | (51) |
| 3.3.4 循环语句.....                 | (58) |
| 3.3.5 跳转语句.....                 | (64) |
| 3.4 数组.....                     | (67) |
| 3.4.1 一维数组.....                 | (67) |
| 3.4.2 多维数组.....                 | (70) |
| 3.4.3 System.arraycopy()方法..... | (73) |
| 习题.....                         | (73) |
| <br>                            |      |
| <b>第 4 章 类、包和接口</b> .....       | (75) |
| 4.1 类、域、方法.....                 | (75) |
| 4.1.1 定义类中的域和方法.....            | (75) |
| 4.1.2 构造方法与对象的创建.....           | (77) |
| 4.1.3 使用对象.....                 | (78) |
| 4.1.4 方法的重载.....                | (79) |
| 4.1.5 this 的使用.....             | (80) |
| 4.2 类的继承.....                   | (81) |
| 4.2.1 派生子类.....                 | (81) |
| 4.2.2 域的继承与隐藏、添加.....           | (82) |
| 4.2.3 方法的继承、覆盖与添加.....          | (83) |
| 4.2.4 super 的使用.....            | (84) |

|              |                     |              |
|--------------|---------------------|--------------|
| 4.2.5        | 父类对象与子类对象的转换        | (85)         |
| 4.3          | 包                   | (88)         |
| 4.3.1        | package 语句          | (88)         |
| 4.3.2        | import 语句           | (88)         |
| 4.3.3        | 编译和运行包中的类           | (89)         |
| 4.3.4        | CLASSPATH           | (90)         |
| 4.4          | 访问控制符               | (90)         |
| 4.4.1        | 成员的访问控制符            | (90)         |
| 4.4.2        | 类的访问控制符             | (95)         |
| 4.4.3        | setor 与 getor       | (96)         |
| 4.4.4        | 构造方法的隐藏             | (96)         |
| 4.5          | 非访问控制符              | (96)         |
| 4.5.1        | static              | (97)         |
| 4.5.2        | final               | (100)        |
| 4.5.3        | abstract            | (101)        |
| 4.5.4        | 其他修饰符               | (103)        |
| 4.5.5        | 一个应用模型——单子          | (104)        |
| 4.6          | 接口                  | (105)        |
| 4.6.1        | 接口的概念               | (105)        |
| 4.6.2        | 定义接口                | (106)        |
| 4.6.3        | 实现接口                | (107)        |
| 4.6.4        | 对接口的引用              | (108)        |
|              | 习题                  | (109)        |
| <b>第 5 章</b> | <b>深入理解 Java 语言</b> | <b>(111)</b> |
| 5.1          | 变量及其传递              | (111)        |
| 5.1.1        | 基本类型变量与引用型变量        | (111)        |
| 5.1.2        | 域变量与局部变量            | (112)        |
| 5.1.3        | 变量的传递               | (113)        |
| 5.1.4        | 变量的返回               | (114)        |
| 5.2          | 多态与虚方法调用            | (114)        |
| 5.2.1        | 上溯造型                | (115)        |
| 5.2.2        | 虚方法调用               | (116)        |
| 5.2.3        | 动态类型确定              | (118)        |
| 5.3          | 对象构造与初始化            | (120)        |
| 5.3.1        | 调用本类或父类的构造方法        | (120)        |
| 5.3.2        | 构造方法的执行过程           | (122)        |
| 5.3.3        | 构建方法内部调用的方法的多态性     | (123)        |
| 5.4          | 对象清除与垃圾回收           | (125)        |

|            |                              |       |
|------------|------------------------------|-------|
| 5.4.1      | 对象的自动清除                      | (125) |
| 5.4.2      | System.gc()方法                | (126) |
| 5.4.3      | finalize()方法                 | (126) |
| 5.5        | 内部类与匿名类                      | (128) |
| 5.5.1      | 内部类                          | (128) |
| 5.5.2      | 方法中的内部类及匿名类                  | (132) |
|            | 习题                           | (134) |
| <b>第6章</b> | <b>异常处理</b>                  | (135) |
| 6.1        | 异常处理                         | (135) |
| 6.1.1      | 异常的概念                        | (135) |
| 6.1.2      | 捕获和处理异常                      | (137) |
| 6.1.3      | 应用举例                         | (140) |
| 6.2        | 创建用户自定义异常类                   | (143) |
|            | 习题                           | (144) |
| <b>第7章</b> | <b>工具类及常用算法</b>              | (146) |
| 7.1        | Java 语言基础类                   | (146) |
| 7.1.1      | Java 基础类库                    | (146) |
| 7.1.2      | Object 类                     | (148) |
| 7.1.3      | 基本数据类型的包装类                   | (152) |
| 7.1.4      | Math 类                       | (153) |
| 7.1.5      | System 类                     | (154) |
| 7.2        | 字符串                          | (155) |
| 7.2.1      | String 类                     | (155) |
| 7.2.2      | StringBuffer 类               | (159) |
| 7.2.3      | StringTokenizer 类            | (161) |
| 7.3        | 集合类                          | (162) |
| 7.3.1      | 集合与 Collection API           | (162) |
| 7.3.2      | Set 接口及 HashSet 类            | (163) |
| 7.3.3      | List 接口及 ArrayList, Vector 类 | (164) |
| 7.3.4      | Iterator 及 Enumeration       | (165) |
| 7.3.5      | Map 接口及 Hashtable 类          | (167) |
| 7.4        | 向量、堆栈、队列                     | (168) |
| 7.4.1      | Vector 向量                    | (168) |
| 7.4.2      | Stack 堆栈                     | (171) |
| 7.4.3      | LinkedList 及队列               | (173) |
| 7.5        | 排序与查找                        | (174) |
| 7.5.1      | Arrays 类                     | (174) |

|              |                        |       |
|--------------|------------------------|-------|
| 7.5.2        | Collections 类          | (176) |
| 7.5.3        | 冒泡排序                   | (178) |
| 7.5.4        | 选择排序                   | (179) |
| 7.5.5        | 快速排序                   | (180) |
| 7.6          | 7.6 遍试、迭代和递归           | (183) |
| 7.6.1        | 遍试                     | (183) |
| 7.6.2        | 迭代                     | (185) |
| 7.6.3        | 递归                     | (188) |
|              | 习题                     | (193) |
| <b>第 8 章</b> | <b>Java 的多线程</b>       | (196) |
| 8.1          | 8.1 线程及其创建             | (196) |
| 8.1.1        | Java 中的线程              | (196) |
| 8.1.2        | 创建线程对象的两种方法            | (197) |
| 8.1.3        | 多线程                    | (199) |
| 8.1.4        | 应用举例                   | (201) |
| 8.2          | 8.2 线程的调度              | (205) |
| 8.2.1        | 线程的状态与生命周期             | (205) |
| 8.2.2        | 线程调度与优先级               | (206) |
| 8.2.3        | 对线程的基本控制               | (208) |
| 8.3          | 8.3 线程的同步与共享           | (211) |
| 8.3.1        | synchronized 关键字       | (211) |
| 8.3.2        | 线程间的同步控制               | (214) |
|              | 习题                     | (219) |
| <b>第 9 章</b> | <b>流、文件及基于文本的应用</b>    | (220) |
| 9.1          | 9.1 流式输入与输出            | (220) |
| 9.1.1        | 字节流与字符流                | (220) |
| 9.1.2        | 节点流和处理流                | (222) |
| 9.1.3        | 标准输入和标准输出              | (225) |
| 9.1.4        | 应用举例                   | (226) |
| 9.2          | 9.2 文件及目录              | (229) |
| 9.2.1        | 文件与目录管理                | (229) |
| 9.2.2        | 文件输入与输出流               | (231) |
| 9.2.3        | RandomAccessFile 类     | (234) |
| 9.3          | 9.3 基于文本的应用            | (235) |
| 9.3.1        | Java Application 命令行参数 | (235) |
| 9.3.2        | 环境参数                   | (236) |
| 9.3.3        | 处理 Deprecated 的 API    | (237) |

|                         |              |
|-------------------------|--------------|
| 习题                      | (239)        |
| <b>第 10 章 图形用户界面</b>    | <b>(240)</b> |
| 10.1 AWT 组件             | (240)        |
| 10.1.1 图形用户界面概述         | (240)        |
| 10.1.2 AWT 组件分类         | (241)        |
| 10.1.3 Component 的方法    | (244)        |
| 10.2 布局管理               | (245)        |
| 10.2.1 FlowLayout       | (245)        |
| 10.2.2 BorderLayout     | (246)        |
| 10.2.3 CardLayout       | (247)        |
| 10.2.4 GridLayout       | (249)        |
| 10.2.5 GridBagLayout    | (250)        |
| 10.2.6 通过嵌套来设定复杂的布局     | (250)        |
| 10.3 事件处理               | (251)        |
| 10.3.1 事件及事件监听器         | (251)        |
| 10.3.2 事件监听器的注册         | (253)        |
| 10.3.3 事件适配器            | (258)        |
| 10.3.4 内部类及匿名类在事件处理中的应用 | (259)        |
| 10.4 常用组件的使用            | (262)        |
| 10.4.1 标签、按钮与动作事件       | (262)        |
| 10.4.2 文本框、文本区域与文本事件    | (264)        |
| 10.4.3 单、复选按钮，列表与选择事件   | (265)        |
| 10.4.4 调整事件与滚动条         | (269)        |
| 10.4.5 鼠标、键盘事件与画布       | (271)        |
| 10.4.6 Frame 与窗口事件      | (273)        |
| 10.4.7 Panel 与容器事件      | (274)        |
| 10.4.8 组件事件、焦点事件与对话框    | (276)        |
| 10.5 绘图、图形和动画           | (277)        |
| 10.5.1 绘制图形             | (277)        |
| 10.5.2 显示文字             | (281)        |
| 10.5.3 控制颜色             | (282)        |
| 10.5.4 显示图像             | (283)        |
| 10.5.5 实现动画效果           | (283)        |
| 10.6 Applet             | (285)        |
| 10.6.1 Applet 的基本工作原理   | (286)        |
| 10.6.2 Applet 类         | (286)        |
| 10.6.3 HTML 文件参数传递      | (288)        |
| 10.6.4 Applet 的局限       | (289)        |

|               |                           |              |
|---------------|---------------------------|--------------|
| 10.7          | SwingGUI 组件 .....         | (295)        |
| 10.7.1        | Swing 的特点 .....           | (295)        |
| 10.7.2        | 几种 Swing 组件介绍 .....       | (296)        |
| 10.8          | 基于 GUI 的应用程序 .....        | (299)        |
| 10.8.1        | 使用可视化设计工具 .....           | (299)        |
| 10.8.2        | 菜单的定义与使用 .....            | (299)        |
| 10.8.3        | 菜单、工具条及对话框的应用 .....       | (303)        |
|               | 习题 .....                  | (308)        |
| <b>第 11 章</b> | <b>网络、多媒体和数据库编程 .....</b> | <b>(310)</b> |
| 11.1          | Java 网络编程 .....           | (310)        |
| 11.1.1        | 使用 URL .....              | (310)        |
| 11.1.2        | 用 Java 实现底层网络通信 .....     | (312)        |
| 11.1.3        | 实现多线程服务器程序 .....          | (314)        |
| 11.1.4        | Java 的 Email 编程 .....     | (323)        |
| 11.2          | 多媒体编程 .....               | (329)        |
| 11.2.1        | 在 Applet 中获取声音与图像 .....   | (329)        |
| 11.2.2        | Java 图像编程 .....           | (332)        |
| 11.2.3        | Java 声音编程 .....           | (334)        |
| 11.3          | Java 数据库编程 .....          | (335)        |
| 11.3.1        | Java 访问数据库的基本步骤 .....     | (335)        |
| 11.3.2        | 使用 JTable 显示数据表 .....     | (339)        |
| 11.4          | J2EE 及 J2ME 简介 .....      | (343)        |
| 11.4.1        | J2EE 简介 .....             | (343)        |
| 11.4.2        | J2ME 简介 .....             | (345)        |
|               | 习题 .....                  | (347)        |
|               | 参考文献 .....                | (348)        |

# 前 言

在程序设计的教学中,选择一种适合的语言是十分重要的。比较多种程序设计语言,笔者认为 Java 具有如下突出的优点。

其一,Java 是面向对象的语言,与现代面向对象的设计与分析的软件工程相一致,也是当前的主流程序设计语言之一。

其二,简单易学。其中的数据类型、数据运算、程序控制结构等基本概念对于任何语言都是一致的;而其语法相对于 C++等语言而言更简单,更容易掌握。

其三,Java 语言就其本身而言支持一些高级特性,如多线程、异常处理、自动垃圾回收等,这些特性使 Java 成为极优秀的语言之一。

最后,Java 具有广泛的用途。Java 具有跨平台的特点,在各种平台上都有应用,它还可以有效地进行数据库、多媒体及网络的程序设计。不仅如此,Java 还与 JavaScript 十分相似,而后者可以广泛用于办公软件、网页设计、网络服务程序等方面。

综上所述,Java 是特别适合于程序设计学习的基础语言。

对于学习者而言,选择一本好的教材至关重要。现在市面上有关 Java 的书不少,但适合于教学、自学的书却不多见。笔者基于多年程序设计语言的教学经验,结合个人的软件开发实践,力图使本书突出以下特色。

1. 对 Java 语言的基础知识,包括数据类型、流程控制、类的封装与继承、多态、虚方法调用、传值调用等进行系统讲解,让学习者知其然,并知其所以然。

2. 对 Java 中的类库中的基本类,包括 Math、字符串、集合进行详细讲解,以利于学习者打下牢固的基础。

3. 对 Java 中的基本应用,包括 I/O、文本界面、图形界面等,精选大量典型而实用的例子,力图使学习者触类旁通,举一反三。

4. 对一些高级应用,如数据库编程、网络编程、多媒体编程等内容,介绍了其概念、原理,以利于学习者能了解 Java 的实际应用及最新发展。

5. 在讲解语言的同时,介绍它所采用的面向对象技术的基础理论、主要原则和思维方法,同时介绍在 Java 软件工程中常用的 UML 工具。

6. 在讲解、举例时充分考虑到各个层次的需要,力求语言简洁,内容循序渐进。同时,考虑到部分读者有参加 Sun 的 SCJP 认证考试的需要,本书提供了较多的习题。

本书在内容安排上,大致可以分为三部分:第一部分介绍了 Java 语言基础,包括数据、控制结构、数组、类、包、对象、接口等;第二部分介绍了 Java 深入知识,包括传值调用、虚方法调用、异常处理、工具类与算法;第三部分是 Java 的应用,包括线程、流式文件、AWT 及 Swing 图形用户界面,以及 Java 在网络、多媒体、数据库等方面的应用。

本教材的内容和组织方式适合作为高等学校各专业的计算机程序设计课程的教材,或者作为计算机技术的培训教材,也可以作为 Java 认证考试的考试用书。

书中存在的缺点和不足,恳请读者批评指正。

唐大仕

2003 年 4 月

于北京大学信息科学技术学院

# 北京大学信息技术系列教材

## 序 言

人类已进入21世纪，科学技术突飞猛进，知识经济初见端倪，特别是信息技术和网络技术的迅速发展和广泛应用，对社会的政治、经济、军事、科技和文化等领域产生越来越深刻的影响，也正在改变着人们的工作、生活、学习和交流方式。信息的获取、处理、交流和应用能力，已经成为人们最重要的能力之一。培养一大批掌握和应用现代信息技术和网络技术的人才，在全球信息化的发展中占据主动地位，不仅是经济和社会发展的需要，也是计算机和信息技术教育者的历史责任。

加入WTO，意味着我国要在同一个网络平台上参与国际竞争，同世界接轨。这对我们既是一个机遇，也是一个挑战。为此我们必须加强全民的信息技术教育，以提高国民的整体素质，抓住国际大环境给我国经济腾飞带来的难得机遇，迎接挑战。

教育部提出，要在全国的中小学中逐步开设信息技术必修课，从小培养获取、分析、处理、发布和应用信息的能力和素养，在条件成熟时，考虑作为普通高校招生考试的科目。国家经贸委也提出，要像抓3年改革和脱困的两大目标那样，把企业管理信息化建设作为新世纪经贸工作的历史性任务抓紧、抓好，推进企业应用计算机管理软件和网络信息技术，用3年左右的时间，在国家重点企业中建立完善的企业信息管理系统。

为了适应这个大的形势，满足各大专院校非计算机专业学生和社会各阶层从事信息技术和急需掌握信息技术人们的需要，我们组织编写了这套《北京大学信息技术系列教材》。目的是让更多的人以最快的速度掌握计算机信息技术，学会运用国际互联网络平台，不断提高自身素质和专业水平，在传统产业升级、实现跨越式发展中更好地展示自己的才能，为祖国的现代化建设服务。

本系列教材包括《计算机信息技术基础》、《计算机网络应用技术》、《办公自动化软件》、《多媒体应用技术》、《网络程序设计——ASP》、《数据库技术——SQL》、《Visual Basic程序设计》、《Visual FoxPro程序设计》、《C++语言程序设计》、《网页制作技术》、《从HTML到XML》、《计算机局域网实用技术》、《Java程序设计》等。随着信息技术的发展和读者的需要，我们还将不断对这一系列教材进行补充或增删，以期形成读者欢迎的动态系列教材。此系列教材可作为大专院校非计算机专业信息技术普及教材，也可供社会各种信息技术培训班选用。

本系列教材具有以下编写特点：

### 1. 适合不同层次的读者选用

此系列教材从内容上讲，跨度较大，从计算机基础知识一直到动态网站制作，这样可以满足不同领域和不同层次的读者需要，读者可以根据自己的水平像吃自助餐一样自主选用。

### 2. 选材超前，出版周期短

目前，计算机图书市场火爆，需求旺盛，但是，选一本合适的教材又非易事。其原因之一一是读者急需使用的高版本软件对应的书上市甚少。造成这种现象的原因有三：一是信息技术发展速度太快；二是选材没有注意超前量；三是出版周期太长。鉴于以上原因，本系列教材在内容上尽量注意超前量，如每一个软件必须选择当前最高版本。例如：动态网站制作我们选择当前流行的ASP技术和SQL网上数据库以及VB编程技术；在保证书稿质量的前提下尽量缩短出版周期。其目的都是为了适应信息技术的飞速发展，满足读者的需要。

### 3. 实用性强

本系列教材的主要对象是非计算机专业人员，因此，在内容上强调实用，尽量不涉及高深的与软件使用无关的理论问题。比如《多媒体应用技术》，作者着重阐述多媒体信息的获取、处理、传输、保存、制作等实用技术，不涉及多媒体的理论问题。又如《计算机局域网实用技术》，作者重点介绍局域网的构架、服务器的安装、各种网上信息服务的建立以及网络安全管理方面的内容，读者可按照书中所讲的内容自己独立构建局域网。

### 4. 充分体现案例教学

在本系列丛书中读者会发现，凡是操作型软件都是以一个案例为主线进行阐述，这是本系列书作者多年来在教学第一线经验的总结。案例教学引人入胜，易理解，易掌握，能使读者举一反三，技术掌握扎实。

### 5. 写作风格通俗易懂

介绍每一个软件开门见山，语言简明扼要，重点突出，难点翔实编写，同一功能决不重复。并在每章中附有习题，有的例题配有光盘，适合自学。

参加本系列教材编写的作者都是在大学从事信息技术课一线教学的中、青年教师，他们都有极强的敬业精神，本系列教材凝聚了他们多年丰富的教学经验和心血。

本系列教材得到了北京大学教育学院教育技术系各位老师和北京大学信息管理系余锦凤教授的支持和帮助，在此表示诚挚的感谢。

由于本系列教材从策划到出版仅仅用了不到一年的时间，编写者又都担负着繁重的教学任务，在时间紧、任务重的情况下，肯定有不少不尽人意之处，诚挚接受广大读者的批评、指正。

蔡翠平

2003年4月于北京大学

# 第 1 章 Java 语言与面向对象的程序设计

Java 语言是当今流行的网络编程语言，它的面向对象、跨平台、分布应用等特点给编程人员带来了一种崭新的计算概念，使 WWW 从最初的单纯提供静态信息发展到现在的提供各种各样的动态服务，产生了巨大的变化。Java 不仅能够编写小应用程序实现嵌入网页的声音和动画功能，而且还能够应用于独立的大中型应用程序，其强大的网络功能能够把整个 Internet 作为一个统一的运行平台，极大地拓展了传统单机或 Client/Server 模式应用程序的外延和内涵。自 1995 年正式问世以来，Java 已经逐步从一种单纯的计算机高级编程语言发展为一种重要的 Internet 平台，并进而引发、带动了 Java 产业的发展壮大，成为当今计算机业界不可忽视的力量和重要的发展潮流与方向。

## 1.1 Java 语言简介

### 1.1.1 Java 语言出现的背景、影响及应用前景

1991 年，SUN MicroSystem 公司的 Jame Gosling, Bill Joe 等人，为在电视机、控制烤面包箱等家用消费类电子产品上进行交互式操作而开发了一个名为 Oak（一种橡树的名字）的软件，但当时并没有引起人们的注意，直到 1994 年下半年，Internet 的迅猛发展，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，一些著名的计算机公司纷纷购买了 Java 语言的使用权，如 Microsoft, IBM, Netscape, Novell, Apple, DEC, SGI 等。因此，Java 的诞生对整个计算机产业产生了深远的影响，可以说，Java 为 Internet 和 WWW 开辟了一个崭新的时代。

Java 对传统的计算模型提出了新的挑战。业界不少人预言：“Java 语言的出现，将会引起一场软件革命。”这是因为传统的软件往往都是与具体的实现环境有关，而 Java 语言能在执行码（二进制码）上兼容，这样，以前所开发的软件就能运行在不同的机器上，只要所用的机器能提供 Java 语言解释器即可。

Java 语言将对未来软件的开发产生影响，可反映在如下几个方面。

(1) 软件的需求分析。可将用户的需求进行动态的、可视化描述，以满足设计者更加直观的要求。Java 语言不受地区、行业、部门、爱好的限制，都可以将用户的需求描述清楚。

(2) 软件的开发方法。由于 Java 语言的面向对象的特性，所以完全可以用面向对象的技术与方法来进行开发，符合最新的软件开发规范的要求。

(3) Java 语言的动态效果。就界面而言，GUI 技术达到动画效果；就数据而言，Java 能根据数据动态地提供信息。

(4) 软件最终产品。用 Java 语言开发的软件可以具有可视化、可听化、可操作化的效果，其多媒体应用也十分广泛。

(5) 其他。使用 Java 语言对开发效益、开发价值都有比较明显的影响。

正如 Java 的创始人之一 James Gosling 所说, Java 不仅仅只是 applets, 它能做任何事情: Java 不仅仅是一种程序设计语言, 更是现代化软件再实现的基础; Java 还是未来新型 OS 的核心; 将会出现 Java 芯片; Java 将构成各种应用软件的开发平台与实现环境, 是人们必不可少的开发工具。

因此, Java 语言有着广泛的应用前景, 例如:

- (1) 所有面向对象的应用开发, 包括面向对象的事件描述、处理和综合等;
- (2) 计算过程的可视化、可操作化的软件的开发;
- (3) 动态画面的设计, 包括图形图像的调用;
- (4) 交互操作的设计 (选择交互、定向交互和控制流程等);
- (5) Internet 的系统管理功能模块, 包括 Web 页面的动态设计、管理和交互操作设计等;
- (6) Intranet (企业内部网) 上的软件开发 (直接面向企业内部用户的软件);
- (7) 与各类数据库连接查询的 SQL 语句实现;
- (8) 其他应用类型的程序。

## 1.1.2 Java 的特点

简单地讲, Java 是定位于网络计算的计算机语言, 它的几乎所有的特点也是围绕这一中心展开并为之服务的, 这些特点使得 Java 语言特别适合用来开发网络上的应用程序。另外, 作为一种问世较晚的语言, Java 也集中体现和充分利用了若干当代软件技术新成果, 如面向对象、多线程等, 这些也都在它的特点中有所反映。Java 的特点如下。

### 1. 简单易学

衍生自 C++ 的 Java 语言, 出于安全稳定性的考虑, 去除了 C++ 中不容易理解和掌握的部分, 如最典型的指针操作等, 降低了学习的难度; 同时 Java 还有一个特点就是它的基本语法部分与 C 语言几乎一模一样。这样, 无论是掌握了 Java 再学 C 语言, 还是已经掌握了 C 语言再来学 Java, 都会感到易于入门。

### 2. 面向对象

Java 是面向对象的编程语言。面向对象技术较好地解决了当今软件开发过程中新出现的种种传统面向过程语言所不能处理的问题, 包括软件开发的规模扩大、升级加快、维护量增大, 以及开发分工日趋细化、专业化和标准化等, 是一种迅速成熟、推广的软件开发方法。面向对象技术的核心是以更接近于人类思维的方式建立计算机逻辑模型, 它利用类和对象的机制将数据与其上的操作封装在一起, 并通过统一的接口与外界交互, 使反映现实世界实体的各个类在程序中能够独立、自治、继承。这种方法非常有利于提高程序的可维护性和可重用性, 大大提高了开发效率和程序的可管理性, 使得面向过程语言难于操纵的大规模软件可以很方便地创建、使用和维护。C++ 也是面向对象的语言, 但是为了与 C 语言兼容, 其中还包含了一些面向过程的成分; Java 去除了 C++ 中非面向过程的部分, 其

程序编写过程就是设计、实现类，定义其属性、行为的过程。

### 3. 平台无关性

如前所述，Java 独特的运行机制使得它具有良好的二进制级的可移植性，利用 Java 语言，开发人员可以编写出与具体平台无关、普遍适用的应用程序，大大降低了开发、维护和管理开销。

### 4. 安全稳定

对网络上应用程序的另一个需求是较高的安全可靠。用户通过网络获取并在本地运行的应用程序必须是可信赖的，不会充当病毒或其他恶意操作的传播者而攻击用户本地的资源；同时它还应该是稳定的，轻易不会产生死机等错误，使得用户乐于使用。Java 特有的机制是其安全性的保障，同时它去除了 C++ 中易造成错误的指针，增加了自动内存管理等措施，保证了 Java 程序运行的可靠性。

### 5. 支持多线程

多线程是当今软件技术的又一重要成果，已成功应用在操作系统、应用开发等多个领域。多线程技术允许同一个程序有两个执行线索，即同时做两件事情，满足了一些复杂软件的需求。Java 不但内置多线程功能，而且提供语言级的多线程支持，即定义了一些用于建立、管理多线程的类和方法，使得开发具有多线程功能的程序变得简单、容易和有效。

### 6. 很好地支持网络编程

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。Java 的小应用程序 (Applet) 是动态、安全、跨平台的网络应用程序。Java Applet 嵌入 HTML 语言，通过主页发布到 Internet。网络用户访问服务器的 Applet 时，这些 Applet 从网络上进行传输，然后在支持 Java 的浏览器中运行。由于 Java 语言的安全机制，用户一旦载入 Applet，就可以放心地生成多媒体的用户界面或完成复杂的计算而不必担心病毒的入侵。虽然 Applet 可以和图像、声音、动画等一样从网络上下载，但它不同于这些多媒体的文件格式，它可以接收用户的输入，动态地进行改变，而不仅仅是动画的显示和声音的播放。

### 7. Java 丰富的类库

Java 提供了大量的类库以满足网络化、多线程、面向对象系统的需要。

- (1) 语言包提供的支持包括字符串处理、多线程处理、例外处理、数学函数处理等，可以用它简单地实现 Java 程序的运行平台。
- (2) 实用程序包提供的支持包括哈希表、堆栈、可变数组、时间和日期等。
- (3) 输入输出包用统一的“流”模型来实现所有格式的 I/O，包括文件系统、网络及输入/输出设备等。
- (4) 低级网络包用于实现 Socket 编程。
- (5) 抽象图形用户接口包实现了不同平台的计算机的图形用户接口部件，包括窗口、

菜单、滚动条和对话框等，使得 Java 可以移植到不同平台的机器。

(6) 网络包支持 Internet 的 TCP/IP 协议，提供了与 Internet 的接口。它支持 URL 连接，WWW 的即时访问，并且简化了用户/服务器模型的程序设计。

Java 的上述种种特性不但能适应网络应用开发的需求，而且还体现了当今软件开发方法的若干新成果和新趋势。在以后的章节里，将结合对 Java 语言的讲解，分别介绍这些软件开发方法。

### 1.1.3 Java 和 C、C++

对于变量声明、参数传递、操作符、流控制等，Java 使用了和 C、C++ 相同的传统，使得熟悉 C、C++ 的程序员能很方便地进行编程。同时，Java 为了实现其简单、健壮、安全等特性，也摒弃了 C 和 C++ 中许多不合理的内容。下面选择性地讲述几点，对于学过 C 语言或 C++ 语言的读者而言，起一个快速参考的作用。对于未学过 C 语言的读者，可以略过此节。

#### (1) 全局变量

Java 程序中，不能在所有类之外定义全局变量，只能通过在一个类中定义公用、静态的变量来实现一个全局变量。Java 对全局变量进行了更好的封装。而在 C 和 C++ 中，依赖于不加封装的全局变量常常会造成系统的崩溃。

#### (2) Goto 语句

Java 不支持 C、C++ 中的 Goto 语句，而是通过异常处理语句 try、catch、finally 等来代替 C、C++ 中用 Goto 来处理遇到错误时跳转的情况，使程序更可读且更结构化。

#### (3) 指针

指针是 C、C++ 中最灵活，也是最容易产生错误的数据类型。由指针所进行的内存地址操作常会造成不可预知的错误，同时通过指针对某个内存地址进行显示类型转换后，可以访问一个 C++ 中的私有成员，从而破坏了安全性，造成系统的崩溃。而 Java 对指针进行完全的控制，程序员不能直接进行任何指针操作，例如，把整数转化为指针，或者通过指针释放某一内存地址等。同时，数组作为类在 Java 中实现，很好地解决了数组访问越界这一在 C、C++ 中不做检查的错误。

#### (4) 内存管理

在 C 中，程序员通过库函数 malloc() 和 free() 来分配和释放内存，C++ 中则通过运算符 new 和 delete 来分配和释放内存。再次释放已释放的内存块或未被分配的内存块，会造成系统的崩溃；同样，忘记释放不再使用的内存块也会逐渐耗尽系统资源。而在 Java 中，所有的数据结构都是对象，通过运算符 new 为它们分配内存。通过 new 得到对象的处理权，而实际分配给对象的内存可能随程序运行而改变，Java 对此自动地进行管理并且进行垃圾收集，有效地防止了由于程序员的误操作而导致的错误，并且更好地利用了系统资源。

#### (5) 数据类型的支持

在 C、C++ 中，对于不同的平台，编译器为简单数据类型，如 int、float 等分别分配不同长度的字节数，例如，int 在 IBM PC 中为 16 位，在 VAX-11 中为 32 位，这导致了代码的不可移植性，但在 Java 中，对于这些数据类型总是分配固定长度的位数，如对 int 型，