

# Java

## 程序设计实用教程

陈勇孝 郎洪 马春龙 主编

清华大学出版社

TP312/2970

2008

# Java

## 程序设计实用教程

陈勇孝 郎洪 马春龙 主编

清华大学出版社

北京

## 内 容 简 介

本书重点介绍 Java 语言面向对象的程序设计方法和用 Java 语言开发实际网络应用程序及操作数据库的方法、步骤,从 Java 语言开发网络程序应用的角度出发,结合一些有实际应用价值的课题,引导读者提出问题 and 解决问题,提高实际应用的能力,激发创造性思维和提高创新能力。本书是郑州大学程序设计语言教学实践改革资助项目。

全书内容包含 16 章。依次分别是第 1 章 Java 开发环境和程序设计初步;第 2 章 Java 程序设计基础;第 3 章程序结构和常用算法;第 4 章 Java 面向对象编程——类和对象;第 5 章 Java 面向对象编程——继承和多态;第 6 章数组、字符串和实用类;第 7 章图形界面程序设计;第 8 章异常处理机制;第 9 章事件处理机制;第 10 章多线程机制;第 11 章图像和图形处理;第 12 章 Applet、声音和动画程序设计;第 13 章输入输出流和文件操作;第 14 章 JDBC 访问数据库技术;第 15 章网络编程——WWW 和 Socket;第 16 章网络编程——RMI、UDP 和 Servlet。全书内容深入浅出,循序渐进,通俗易懂。

本书可作为计算机应用专业本科生、研究生的教材,也可供从事计算机应用和网络程序开发的各类人员使用。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

## 图书在版编目(CIP)数据

Java 程序设计实用教程/陈勇孝,郎洪,马春龙主编. —北京:清华大学出版社,2008.7  
ISBN 978-7-302-17349-6

I. J… II. ①陈… ②郎… ③马… III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 052216 号

责任编辑:陈国新 陈志辉

责任校对:白 蕾

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京国马印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:26.75 字 数:650 千字

版 次:2008 年 7 月第 1 版 印 次:2008 年 7 月第 1 次印刷

印 数:1~3000

定 价:39.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:026047-01

# 前 言

Java 语言自问世以来,随着网络的快速发展,已逐渐成为网络上最受欢迎的面向对象的程序设计语言。Java 是一种与平台无关的、安全可靠的、真正的面向对象的网络程序设计语言。正如微软总裁比尔·盖茨所说:“Java 是长时间以来最卓越的程序设计语言”,并因此确定了微软的整个软件开发的战略从 PC 单机时代向着以网络为中心的计算时代转移。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系统都要用 Java 语言来改写。工业界不少人预言:“Java 语言的出现,将会引起一场软件革命。”由于 Java 语言符合了互联网时代的发展要求,使各类软件开发人员能更容易地选择多种平台来使用 Java 语言编程,为 Java 语言的应用开拓了极为广阔的前景,使它获得了巨大的成功。

本书以 J2SDK\_1.5.02 为蓝本,根据作者多年来在各种层次本科生和 Java 认证培训教学中的讲授经验提炼而成。重点介绍 Java 语言面向对象的程序设计方法和用 Java 语言开发实际网络应用程序及操作数据库的方法步骤,并从 Java 语言实际开发网络程序应用的角度出发,结合一些有实际应用价值的课题,随着课题内容的展开不断地完善和实现实用程序,目的在于引导读者提出问题和解决问题,提高实际应用的能力,激发创造性思维,提高创新能力。但由于 Java 语言类库庞大、内容太丰富,作为教材又不能面面俱到,因此编者本着加强入门训练,重在掌握方法,提倡应用创新的原则,力求在体系结构、编写风格和内容选材上把握以下几点:

(1) 重点介绍 Java 语言中的精华部分,以面向对象的程序设计方法为主线,展示用 Java 开发实际应用程序的方法和步骤。

(2) 从特定的案例入手,全部讲授内容始终围绕着几个缩小了的特定课题不断地完善程序代码。教材中对 Java 语言不是简单地进行内容罗列,更没有仅为了讲解而给出成段的代码,而是从实用的角度力图提供简洁而完整的示例,读者可在此示例基础上进一步完善,从而提高创新能力。

(3) 书中的数据库操作、客户机/服务器的网络应用等应用课题都是随着内容的展开由浅入深地完善,请读者一定要加强训练,以便通过本书的学习,达到能熟练掌握面向对象程序设计的水平。

全书内容共 16 章,其中第 1~3 章讲述 Java 基础知识;第 4、5 章讲述面向对象程序设计的基本知识,第 6 章对程序设计中的实用类库进行介绍;第 7~10 章对图形界面程序设计、异常处理机制、事件处理机制、多线程机制进行介绍;第 11、12 章讲述图像和图形处理,Applet、声音和动画程序设计实际编程时的应用方法和技巧;第 13、14 章主要讨论对文件

的操作和访问数据库的方法;第15、16章讨论Java网络编程技术。

本书由陈勇孝组织编写。陈勇孝编写了第1章、第9章到第12章和第14章,郎洪编写了第8章、第13章、第15章和第16章,马春龙编写了第4章和第5章,郑明编写了第2章和第3章,孙立贤编写了第7章,黄文君编写了第6章。陈勇孝、郎洪共同负责全书内容安排。

本书可作为计算机应用专业本科生、研究生的教材,也可供从事计算机应用和网络程序开发的各类人员使用。

作者向出版社提供了本书所有程序的源代码,如需要,可与作者(ieyxchen@zzu.edu.cn)或出版社联系。

由于作者水平所限,书中难免有不当之处,还望读者指正。

编 者

2008年4月



# 录



|                                                 |    |
|-------------------------------------------------|----|
| 第 1 章 Java 开发环境和程序设计初步 .....                    | 1  |
| 1.1 Java 语言发展历史 .....                           | 1  |
| 1.1.1 Java 语言产生的背景 .....                        | 1  |
| 1.1.2 互联网推动了 Java 语言的发展 .....                   | 2  |
| 1.2 Java 语言的特点 .....                            | 4  |
| 1.3 Java 的工作原理 .....                            | 5  |
| 1.3.1 Java 虚拟机 .....                            | 5  |
| 1.3.2 内存自动回收机制 .....                            | 5  |
| 1.3.3 代码安全性检查机制 .....                           | 6  |
| 1.4 Java 运行环境 JDK .....                         | 6  |
| 1.4.1 Java 平台 .....                             | 6  |
| 1.4.2 建立 Java 开发环境 .....                        | 7  |
| 1.4.3 几个开发工具的使用 .....                           | 8  |
| 1.5 构建 Java 程序 .....                            | 9  |
| 1.5.1 第一个应用程序 HelloWorld .....                  | 9  |
| 1.5.2 第一个 Java Applet 程序 HelloWorldApplet ..... | 12 |
| 1.6 RAD(快速开发)工具 JCreator .....                  | 14 |
| 第 2 章 Java 程序设计基础 .....                         | 18 |
| 2.1 标识符与关键字 .....                               | 18 |
| 2.1.1 标识符 .....                                 | 18 |
| 2.1.2 关键字 .....                                 | 19 |
| 2.2 基本数据类型、变量和常量 .....                          | 19 |
| 2.2.1 数据类型 .....                                | 19 |
| 2.2.2 变量 .....                                  | 19 |
| 2.2.3 常量和字面量 .....                              | 22 |
| 2.2.4 基本数据类型间的转换 .....                          | 23 |
| 2.2.5 基本数据类型的封装 .....                           | 24 |

|                                    |           |
|------------------------------------|-----------|
| 2.3 操作运算符及表达式 .....                | 26        |
| 2.3.1 算术运算符 .....                  | 26        |
| 2.3.2 逻辑运算符 .....                  | 27        |
| 2.3.3 关系运算符 .....                  | 28        |
| 2.3.4 位运算 .....                    | 29        |
| 2.3.5 其他运算符 .....                  | 30        |
| 2.3.6 运算符的优先级 .....                | 32        |
| 2.4 Java 程序规范 .....                | 32        |
| 2.4.1 Java 源程序结构 .....             | 32        |
| 2.4.2 Java 编程规范 .....              | 34        |
| 2.4.3 Java 帮助文档 .....              | 34        |
| 2.4.4 Java 注释 .....                | 34        |
| 2.5 应用举例 .....                     | 36        |
| 2.5.1 字符界面应用程序 .....               | 36        |
| 2.5.2 GUI 界面应用程序 .....             | 37        |
| 2.5.3 Applet 小应用程序 .....           | 39        |
| <b>第3章 程序结构和常用算法 .....</b>         | <b>41</b> |
| 3.1 分支控制结构 .....                   | 41        |
| 3.1.1 if 结构 .....                  | 41        |
| 3.1.2 switch 结构 .....              | 44        |
| 3.2 循环(或重复)结构 .....                | 48        |
| 3.2.1 for 结构 .....                 | 48        |
| 3.2.2 while 结构 .....               | 51        |
| 3.2.3 do-while 结构 .....            | 52        |
| 3.3 其他辅助语句 .....                   | 53        |
| 3.3.1 break 语句 .....               | 53        |
| 3.3.2 return 语句 .....              | 54        |
| 3.3.3 continue 语句 .....            | 55        |
| 3.4 应用举例 .....                     | 56        |
| <b>第4章 Java 面向对象编程——类和对象 .....</b> | <b>62</b> |
| 4.1 面向对象的基本概念 .....                | 62        |
| 4.1.1 对象、类和消息 .....                | 63        |
| 4.1.2 OOP 的特点 .....                | 64        |
| 4.2 类的定义 .....                     | 66        |
| 4.2.1 Java 类的逻辑结构 .....            | 66        |

|            |                           |           |
|------------|---------------------------|-----------|
| 4.2.2      | 类的定义                      | 67        |
| 4.2.3      | 成员变量                      | 70        |
| 4.3        | 方法                        | 72        |
| 4.3.1      | 方法的定义                     | 72        |
| 4.3.2      | 方法的调用                     | 73        |
| 4.3.3      | 方法参数的传递方式                 | 75        |
| 4.4        | 构造函数                      | 78        |
| 4.4.1      | 无参数的构造函数                  | 79        |
| 4.4.2      | 带参数的构造函数                  | 80        |
| 4.5        | 对象(类的实例化)                 | 81        |
| 4.5.1      | 对象的生成                     | 81        |
| 4.5.2      | 对象的引用                     | 83        |
| 4.5.3      | 对象的清除                     | 84        |
| 4.5.4      | 对象的内存模型                   | 84        |
| 4.6        | 实例成员和类成员                  | 86        |
| 4.6.1      | 实例成员                      | 86        |
| 4.6.2      | 类成员(静态变量、静态方法)            | 87        |
| 4.7        | 应用程序的命令行参数                | 90        |
| 4.8        | 应用举例                      | 92        |
| <b>第5章</b> | <b>Java 面向对象编程——继承和多态</b> | <b>95</b> |
| 5.1        | 类的继承                      | 95        |
| 5.1.1      | 类的继承的实现                   | 95        |
| 5.1.2      | 方法的重载                     | 97        |
| 5.1.3      | 方法的覆盖                     | 99        |
| 5.1.4      | 构造函数的重载                   | 103       |
| 5.2        | 抽象类                       | 104       |
| 5.2.1      | 抽象类                       | 105       |
| 5.2.2      | 抽象方法                      | 106       |
| 5.3        | 最终类                       | 109       |
| 5.3.1      | 最终类                       | 109       |
| 5.3.2      | 最终方法                      | 109       |
| 5.3.3      | 最终变量                      | 110       |
| 5.4        | 接口                        | 110       |
| 5.4.1      | 接口的基本概念                   | 110       |
| 5.4.2      | 接口的声明(定义)                 | 111       |
| 5.4.3      | 接口的实现                     | 112       |



|            |                                 |            |
|------------|---------------------------------|------------|
| 5.5        | 包 .....                         | 113        |
| 5.5.1      | 创建包 .....                       | 114        |
| 5.5.2      | 引用包中的类 .....                    | 116        |
| 5.6        | 访问控制(限定)符 .....                 | 119        |
| 5.6.1      | 类的访问控制符 .....                   | 119        |
| 5.6.2      | 方法与变量的访问控制符 .....               | 119        |
| 5.7        | 应用举例 .....                      | 120        |
| <b>第6章</b> | <b>数组、字符串和实用类 .....</b>         | <b>123</b> |
| 6.1        | 数组 .....                        | 123        |
| 6.1.1      | 一维数组 .....                      | 123        |
| 6.1.2      | 二维数组 .....                      | 126        |
| 6.2        | 字符串类 .....                      | 128        |
| 6.2.1      | String 类 .....                  | 129        |
| 6.2.2      | StringBuffer 类 .....            | 132        |
| 6.2.3      | StringTokenizer 类 .....         | 135        |
| 6.3        | 数组类和向量类 .....                   | 136        |
| 6.3.1      | 数组 Arrays 类 .....               | 136        |
| 6.3.2      | 向量类 Vector .....                | 137        |
| 6.4        | Java 语言基础类库(java. lang 包) ..... | 139        |
| 6.4.1      | Java 类库介绍 .....                 | 139        |
| 6.4.2      | Java 语言核心类库 .....               | 140        |
| 6.4.3      | 实用类库 java. util .....           | 149        |
| 6.5        | 应用举例 .....                      | 154        |
| <b>第7章</b> | <b>图形界面程序设计 .....</b>           | <b>160</b> |
| 7.1        | 图形用户界面构成 .....                  | 160        |
| 7.2        | 容器组件 .....                      | 162        |
| 7.3        | AWT 组件库 .....                   | 166        |
| 7.4        | 菜单程序设计 .....                    | 169        |
| 7.5        | 组件的组织——布局管理器 .....              | 171        |
| 7.5.1      | BorderLayout .....              | 171        |
| 7.5.2      | FlowLayout .....                | 173        |
| 7.5.3      | CardLayout .....                | 174        |
| 7.5.4      | GridLayout .....                | 176        |
| * 7.5.5    | GridBagLayout .....             | 177        |
| 7.5.6      | null 布局 .....                   | 178        |

|               |                                   |            |
|---------------|-----------------------------------|------------|
| 7.6           | Swing 组件简介 .....                  | 179        |
| 7.6.1         | Swing 概述 .....                    | 179        |
| 7.6.2         | Swing 容器 .....                    | 180        |
| 7.6.3         | Swing 组件库 .....                   | 183        |
| 7.7           | 应用举例 .....                        | 188        |
| <b>第 8 章</b>  | <b>异常处理机制 .....</b>               | <b>195</b> |
| 8.1           | 传统的处理方式 .....                     | 195        |
| 8.2           | Java 的异常处理机制 .....                | 197        |
| 8.3           | Java 异常类 .....                    | 200        |
| 8.4           | 异常的处理 .....                       | 204        |
| 8.4.1         | 异常的抛出(throw 语句, throws 关键字) ..... | 204        |
| 8.4.2         | 异常的捕获 .....                       | 206        |
| 8.4.3         | 处理多个异常 .....                      | 210        |
| 8.5           | 用户定义的异常 .....                     | 212        |
| 8.6           | 应用举例 .....                        | 213        |
| <b>第 9 章</b>  | <b>事件处理机制 .....</b>               | <b>218</b> |
| 9.1           | 事件处理模型 .....                      | 218        |
| 9.2           | 事件类 .....                         | 221        |
| 9.3           | 事件监听器接口 .....                     | 224        |
| 9.4           | 适配器 .....                         | 227        |
| 9.5           | 内部类和匿名类 .....                     | 229        |
| 9.5.1         | 内部类 .....                         | 229        |
| 9.5.2         | 匿名类 .....                         | 231        |
| 9.6           | 应用举例 .....                        | 233        |
| <b>第 10 章</b> | <b>多线程机制 .....</b>                | <b>238</b> |
| 10.1          | 多线程的概念 .....                      | 238        |
| 10.1.1        | 进程和线程的概念 .....                    | 238        |
| 10.1.2        | 线程和进程的关系 .....                    | 239        |
| 10.2          | 多线程机制 .....                       | 242        |
| 10.2.1        | 线程接口和线程类 .....                    | 242        |
| 10.2.2        | 线程的状态(生命周期) .....                 | 246        |
| 10.3          | 多线程程序设计 .....                     | 250        |
| 10.4          | 线程的调度 .....                       | 253        |
| 10.5          | 线程的同步 .....                       | 255        |

|               |                               |            |
|---------------|-------------------------------|------------|
| 10.6          | 应用举例 .....                    | 258        |
| <b>第 11 章</b> | <b>图像和图形处理 .....</b>          | <b>261</b> |
| 11.1          | 图像的显示 .....                   | 261        |
| 11.1.1        | 图像格式 .....                    | 261        |
| 11.1.2        | 图像文件的权限 .....                 | 262        |
| 11.1.3        | 图像的显示 .....                   | 264        |
| 11.2          | 图形绘制 .....                    | 266        |
| 11.2.1        | 简单图形的绘制 .....                 | 266        |
| 11.2.2        | 使用颜色 .....                    | 271        |
| 11.2.3        | 绘图模式 .....                    | 272        |
| 11.2.4        | 画笔的样式 .....                   | 273        |
| 11.3          | 应用举例 .....                    | 276        |
| <b>第 12 章</b> | <b>Applet、声音和动画程序设计 .....</b> | <b>279</b> |
| 12.1          | Applet 小应用程序 .....            | 279        |
| 12.1.1        | Applet 程序工作机理 .....           | 279        |
| 12.1.2        | Applet 结构 .....               | 281        |
| 12.1.3        | Web 页与 Applet 标记 .....        | 283        |
| 12.1.4        | HTML 向 Applet 传递参数 .....      | 285        |
| 12.1.5        | Applet 中页面的重定向 .....          | 287        |
| 12.1.6        | Applet 之间的通信 .....            | 288        |
| 12.2          | 声音 .....                      | 291        |
| 12.2.1        | 在 Applet 中播放音频 .....          | 291        |
| 12.2.2        | 在 Application 中播放音频 .....     | 293        |
| 12.3          | 动画 .....                      | 295        |
| 12.3.1        | 使用 Timer 类 .....              | 295        |
| 12.3.2        | 使用线程 .....                    | 297        |
| 12.4          | 应用举例 .....                    | 298        |
| <b>第 13 章</b> | <b>输入输出流和文件操作 .....</b>       | <b>302</b> |
| 13.1          | 输入输出流 .....                   | 302        |
| 13.1.1        | 标准输入输出 .....                  | 303        |
| 13.1.2        | 流的概念 .....                    | 304        |
| 13.1.3        | 流的类型 .....                    | 304        |
| 13.1.4        | 过滤器 .....                     | 305        |
| 13.2          | 字节流 .....                     | 306        |

|                                             |                                |            |
|---------------------------------------------|--------------------------------|------------|
| 13.2.1                                      | 文件输入流 .....                    | 308        |
| 13.2.2                                      | 文件输出流 .....                    | 310        |
| 13.3                                        | 字符流 .....                      | 311        |
| 13.3.1                                      | 读文本文件 .....                    | 311        |
| 13.3.2                                      | 写文本文件 .....                    | 313        |
| 13.4                                        | 文件对象 .....                     | 315        |
| 13.4.1                                      | File 类 .....                   | 315        |
| 13.4.2                                      | 文件对象操作实例 .....                 | 316        |
| 13.5                                        | 随机文件 .....                     | 317        |
| 13.6                                        | 对象的串行化 .....                   | 322        |
| 13.7                                        | 应用举例 .....                     | 324        |
| <b>第 14 章 JDBC 访问数据库技术 .....</b>            |                                | <b>327</b> |
| 14.1                                        | JDBC 技术与 ODBC .....            | 327        |
| 14.1.1                                      | ODBC 技术简介 .....                | 328        |
| 14.1.2                                      | JDBC 简介 .....                  | 328        |
| 14.2                                        | SQL 语句简介 .....                 | 330        |
| 14.3                                        | JDBC 中的主要对象和接口 .....           | 332        |
| 14.4                                        | 操作数据库 .....                    | 341        |
| 14.5                                        | 应用实例 .....                     | 346        |
| <b>第 15 章 网络编程——WWW 和 Socket .....</b>      |                                | <b>354</b> |
| 15.1                                        | 网络知识概述 .....                   | 354        |
| 15.1.1                                      | 网络概述 .....                     | 354        |
| 15.1.2                                      | 网络基本概念 .....                   | 355        |
| 15.2                                        | 基于 URL 对象的网络编程 .....           | 358        |
| 15.2.1                                      | 使用 URL 对象读取 WWW 网络资源 .....     | 358        |
| 15.2.2                                      | 使用 InetAddress 对象解析 URL .....  | 360        |
| 15.2.3                                      | 通过 URLConnection 类连接 WWW ..... | 361        |
| 15.3                                        | 基于 Socket(套接字)的网络编程 .....      | 362        |
| 15.3.1                                      | 客户端套接字 Socket 类 .....          | 362        |
| 15.3.2                                      | 服务器端套接字 ServerSocket 类 .....   | 365        |
| 15.4                                        | 应用举例 .....                     | 369        |
| <b>第 16 章 网络编程——RMI、UDP 和 Servlet .....</b> |                                | <b>375</b> |
| 16.1                                        | 远程方法调用 RMI .....               | 375        |
| 16.1.1                                      | RMI 简介 .....                   | 375        |

|     |                               |     |
|-----|-------------------------------|-----|
| 808 | 16.1.2 RMI 远程接口 .....         | 377 |
| 810 | 16.1.3 RMI 服务器 .....          | 378 |
| 811 | 16.1.4 RMI 客户类 .....          | 380 |
| 818 | 16.1.5 RMI 的分布执行 .....        | 384 |
| 818 | 16.2 数据报文 .....               | 385 |
| 818 | 16.2.1 UDP 简介 .....           | 385 |
| 818 | 16.2.2 DatagramSocket 类 ..... | 386 |
| 818 | 16.2.3 DatagramPacke 类 .....  | 386 |
| 817 | 16.2.4 广播通信中的应用 .....         | 390 |
| 838 | 16.3 Java Servlet .....       | 393 |
| 831 | 16.3.1 Servlet 简介 .....       | 393 |
| 838 | 16.3.2 构建 Servlet .....       | 394 |
| 838 | 思考练习题 .....                   | 399 |
| 838 | 参考文献 .....                    | 416 |



# 第 1 章 Java 开发环境和 程序设计初步

通过本章的学习,使读者初步了解 Java 语言产生的背景历史和工作原理,掌握 Java 语言作为一门面向对象编程语言的基本编程思路,初步接触 Java 程序的两种方式: Java Application 和 Java Applet,掌握 Java 程序的结构,学习阅读 Java 帮助文档,并安装 Java 开发环境和帮助文档。最后还要学会编写 Java 程序的开发工具的使用。

## 【难点与重点】

Java 虚拟机

无用内存自动回收器

Java 语言的特点

Java 运行环境 JDK 安装及配置

Java 开发工具的使用

构建 Java 的方法步骤

快速开发工具的使用

## 【学习目标】

了解 Java 语言产生的背景和发展过程以及 Java 的工作机理,掌握 Java 运行平台的下载、安装、环境的配置方法和 JDK 自带的几个工具的使用,初步认识 Java 语言作为一门面向对象编程语言的基本编程思路,熟悉 Java 程序的两种方式 Java Application 和 Java Applet 的结构。最后掌握 Java 开发常用工具的使用。

## 1.1 Java 语言发展历史

### 1.1.1 Java 语言产生的背景

Java 语言产生于 C++ 语言之后,是完全的面向对象的编程语言,充分吸取了 C++ 语言的优点,采用了程序员所熟悉的 C 和 C++ 语言的许多语法,同时又去掉了 C++ 语言中指针、内存申请和释放等影响程序健壮性的部分,可以说 Java 语言是站在 C++ 语言这个“巨人的肩膀上”前进的语言。

任何事物的产生既有必然的原因也有偶然的因素,Java 语言的出现也验证了这一点。1991 年 4 月,美国 Sun Microsystems 公司的一组人员开始了一个代号为“Green”的工程,Green 工程的目标就是开发出一个适用于家用电器控制的系统。项目组的员工很快意识

到,在家用电器中使用的处理器没有统一的标准,消费类电子产品种类繁多,包括 PDA、机顶盒、手机等,即使是同一类消费电子产品所采用的处理器芯片和操作系统也不相同,也存在着跨平台的问题。为了简化开发工作,他们需要一个与平台无关的开发环境。

Sun Microsystems 公司的 James Gosling 起初试图扩展 C++ 语言,但是研究表明,对于消费类电子产品而言,C++ 语言过于复杂和庞大,并不适用,安全性也并不令人满意,这需要花费很多的精力,并且效果不佳。于是,James Gosling 和 Bill Joy 领导的研究小组就着手设计和开发出一种称为 Oak 的语言,传说将其命名为 Oak 是因为在他们设计该语言的结构时,窗外的一棵橡树(oak)映入了他们的眼帘。后来这个名字又不得不放弃,因为较早的一种语言也用了这个名字。

经过很长一段时间的构思后,在一次小组成员去附近的一家咖啡馆时,灵感就来了,取名 Java 语言。Java 是印尼的一个小岛,盛产咖啡,而程序员往往喜欢喝咖啡,因此直到现在“一杯飘香的咖啡”仍然作为该语言的标志,目前 Java 这杯咖啡已经飘香于世界各地。

### 1.1.2 互联网推动了 Java 语言的发展

其实在当时,Green 工程完成后 Java 被束之高阁。1993 年 Green 小组加入到 Sun Microsystems 公司的一个下属公司 Firstperson,投标 time-Warner 公司的一个电视研究项目并失败,1994 年与 3DO 公司的生意同样失败,一半成员去了 Sun Interactive 研究数字视频服务器,另一半到 Sun 研究多媒体和网络计算。

正当 Green 小组遭受这些挫折的时候,直到 1994 年下半年,Internet 的迅猛发展,环球信息网 WWW 的快速增长,WWW 主宰了 Internet,促进了 Java 语言研制的进展,Sun Microsystems 意识到了 Java 的技术潜能,推出了完全用 Java 编写的 Internet 上新型浏览器 HotJava,使得它逐渐成为 Internet 上受欢迎的开发与编程语言。

Sun 公司绝没想到本想用于开发消费类电子产品的编程语言率先在网络中得到了广泛应用,正是因为 Java 语言在设计目标上的正确性使得它得到了业界的广泛认同。

1995 年,美国 Sun Microsystems 公司正式向 IT 业界推出了 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,当时以 Web 为主要形式的互联网正在迅猛发展,Java 语言的出现迅速引起所有程序员和软件公司的极大关注,程序员们纷纷尝试用 Java 语言编写网络应用程序,并利用网络把程序发布到世界各地来运行。包括 IBM、Oracle、Microsoft、Netscape、Apple、SGI 等大公司纷纷与 Sun Microsystems 公司签订合同,授权使用 Java 平台技术。

我们知道,在 PC 中 Windows 系统下编写的程序不做修改就直接拿到 UNIX 系统下运行是不可能的,因为程序最终必须转换成为计算机硬件的机器指令来执行,专门为某种计算机硬件和操作系统编写的程序是不能够直接放到另外的计算机硬件上执行的,至少要做移植工作。要想让程序能够在不同的计算机上运行,就要求程序设计语言是能够跨越各种软件和硬件平台的,而 Java 满足了这一需求。

在经历了以大型机为代表的集中计算模式和以 PC 机为代表的分散计算模式之后,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,即计算机的类型和操作系统是不一样的,例如 Sun 工作站的硬件是 SPARC 体系,软件

是 UNIX 中的 Solaris 操作系统,而 PC 机的硬件是 Intel 体系,操作系统是 Windows 或者 Linux,因此相应的编程语言基本上只是适用于单机系统,例如,COBOL、FORTRAN、C、C++ 等;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的各种计算机上正常运行,Java 就是在这种需求下应运而生的。正是因为 Java 语言符合了互联网时代的发展要求,才使它获得了巨大的成功。

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之一(计算机类就此一项入选),随之出现了大量用 Java 编写的软件产品,受到工业界的重视与好评,认为“Java 是 20 世纪 80 年代以来计算机界的一件大事”,微软总裁比尔·盖茨说:“Java 是长时间以来最卓越的程序设计语言”,并确定微软的整个软件开发战略从 PC 单机时代向着以网络为中心的计算机时代转移,而购买 Java 则是他的重大战略决策的实施部署。

使用 Java 已成大势所趋,Microsoft 和 IBM 两大公司都计划在 Internet 上销售用 Java 编写的软件。SunSoft 公司(这是 Sun 下属的一个子公司),将通过颁发许可证的办法来许可各家公司把 Java 虚拟机和 Java 的 Applets 类库嵌入他们开发的操作系统,这样各类开发人员就能更容易地选择多种平台来使用 Java 语言编程,不同的用户也就可以脱离 Web 浏览器来运行 Java 应用程序,这无疑是很受广大用户欢迎的,也为 Java 语言的应用开拓了极为广阔的前景。

另外 Java 不仅能编制动态的应用软件,而且能编制完整的成套桌面应用软件,将来还会提供更多的能帮助编制动态应用软件的模块,显然,这也是为 Java 的应用提供了有力的例证。

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。工业界不少人预言:“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,换了一个环境就需要作一番改动,耗时费力,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。目前许多大学纷纷开设 Java 课程,Java 正逐步成为世界上程序员用得最多的编程语言。

Java 语言将对未来软件的开发产生影响,可从如下几个方面考虑:

(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,以直观的形式满足设计者要求。虽然用户的需求是各种各样的,受地区、行业、部门、爱好的影响,但都可以用 Java 语言描述清楚。

(2) 软件的开发方法:由于 Java 语言是面向对象的,所以完全可以用 OO 的技术与方法来开发,这是符合最新的软件开发规范要求的。

(3) Java 语言的动画效果远比 GUI 技术更加逼真,尤其是利用 WWW 提供的巨大动画资源空间,可以共享全世界的动画资源。

(4) 软件最终产品:用 Java 语言开发的软件可视、可听、可操作,这要比电视、电影的效果更为理想,因为它可以做到“即时、交互、动画与动作”,要它停就停,要它继续就继续,而这是在电影与电视播放过程中难以做到的。

(5) 其他:使用 Java 语言对开发效益、开发价值都有比较明显的益处。

## 1.2 Java 语言的特点

Java 到底是什么样的语言呢?

### 1. 简单并面向对象

Java 的简单性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。和所有的新一代的程序设计语言一样,Java 也采用了面向对象技术并且更加彻底,所有的 Java Application 和 Applet 均是对象,封装实现了模块化和信息隐藏,继承实现了代码的复用,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、多重继承的复杂概念,而采用了单一继承、类强制转换、多线程、引用(非指针)等方式。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,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。

### 2. 鲁棒并安全

Java 语言在编译及运行程序时,都要进行严格的检查,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 程序中不采用地址计算的方法而用指针访问内存单元,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,这样就可以在编译中避免数组越界的发生。内存自动回收机制也增加了 Java 的鲁棒性。

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 极高的鲁棒性也增强了 Java 的安全性。

### 3. 结构中立并可以移植

网络上充满了各种不同类型的机器和操作系统,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格式文件。任何种类的计算机,只要在其处理器和操作系统上有 Java 运行时环境,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变得不可能。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。