

21 世纪大学计算机专业教材

Java

程序设计

王志文 夏 秦 李平均 编著
陆丽娜 审



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

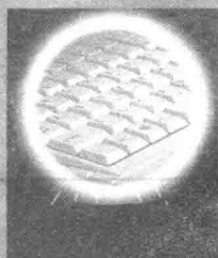
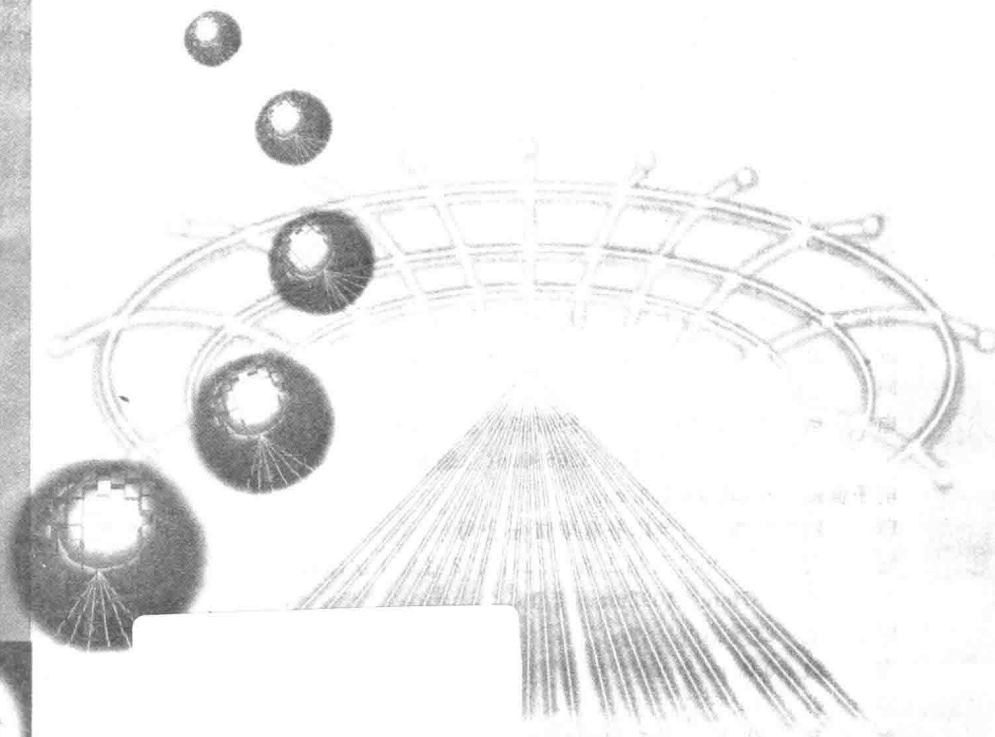
TP312
1257

21世纪大学计算机专业教材

Java

程序设计

王志文 夏 秦 李平均 编著
陆丽娜 审



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

· 西安 ·

内 容 提 要

本书阐述了 Java 面向对象程序设计方法,共分 18 章,主要包括以下内容:Java 开发环境、Java 语言基础、Java 事件处理和异常机制、Java 类的创建与应用、Java Applet 基础、图形用户界面设计与布局管理器、输入/输出、网络编程、JDBC 数据库编程以及 RMI 与 CORBA 分布式编程技术。每一章都明确指出了应该掌握的重要内容,并附有课后练习题。

本书的特点是概念清晰、论述严谨、内容新颖、图文并茂、例程丰富,既重视基本原理和基本概念的阐述,又力图反映出 Java 语言的一些最新发展。本书可以作为高等院校计算机及相关专业的研究生、本科生教材,并可供各行各业从事计算机应用,特别是从事 Internet 网络应用编程的程序员使用。

图书在版编目(CIP)数据

Java 程序设计/王志文,夏秦,李平均编著. —西安:西安交通大学出版社,2004. 2

(21 世纪大学计算机专业教材)

ISBN 7-5605-1803-6

I. J… II. ①王… ②夏… ③李… 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 119314 号

- 书 名 Java 程序设计
编 著 王志文 夏秦 李平均
责任编辑 贺峰涛 屈晓燕
文字编辑 俞晓红
出版发行 西安交通大学出版社
地 址 西安市兴庆南路 25 号(邮编:710049)
网 址 <http://unit.xjtu.edu.cn/unit/jtupress>
电 话 (029)82668357 82667874(发行部)
(029)82668315 82669096(总编部)
电子信箱 eibooks@163.com
印 刷 陕西宝石兰印务有限责任公司
版 次 2004 年 2 月第 1 版 2004 年 2 月第 1 次印刷
开 本 787mm×1092mm 1/16
印 张 21.625
印 数 0 001~3 000
字 数 515 千字
书 号 ISBN7-5605-1803-6/TP·361
定 价 28.50 元

版权所有 侵权必究

前 言

Java 是 Sun Microsystems 公司开发的强有力的新型程序设计语言。Java 语言之所以著名,不仅因为 Java Applets 可以在 Web 页面中运行,而且因为它确实是一种强有力的易于使用的面向对象程序设计语言。Java 语言可以处理许多常见的、但却相当复杂的问题,而这些问题是程序员在开发应用程序时经常碰到的。Java 通过它所提供的线程类来支持多线程程序设计,它还可以在后台执行垃圾回收功能,自动释放不再被使用的内存空间。Java 应用程序设计接口(Application Programming Interface,简称 API)包含在 Sun 公司所提供的 Java 开发工具(Developer's Kit)中,这些接口为程序员在编制复杂的 Internet 应用程序时,提供了对所需工具的平台无关性访问(platform-independent access)。

Java 使得平台无关性处理的理想成为现实。Java Applets 可以运行在任何可以运行 Java 型 Web 浏览器的机器上,而单个的 Java 程序则可以被编译为与平台无关的字节代码,然后,这些字节代码可以在任何配有 Java 解释器的机器上运行。可以说,Java 是第一种真正实现平台无关处理的流行高级程序设计语言。

为适应当前 Internet 的迅猛发展及相关专业人士学习 Java 语言的需要,尤其是根据高等院校的本科学生教学要求,结合笔者多年对 Java 语言的跟踪以及自身的应用开发体会,特编写《Java 程序设计》一书。本书共分 18 章,涉及的内容包括 Java 语言的发展、开发环境、基本语法、面向对象编程、网络编程、I/O 操作、JDBC 数据库编程以及 RMI 和 CORBA 等分布式编程技术。

编写教材最难处理的就是内容的取舍。Internet 网络技术的飞速发展以及软件开发模型的层次化走向使得 Java 语言也在不断地更新。在非常有限的篇幅中,应当将哪些最为重要的内容交给学生?经验证明,最重要的是要在教材中把该课程涉及的基本原理讲述清楚。尽管理论联系实际非常必要,但教材不能当作工程实践的手册指南使用。新的但尚未成熟的内容不宜写入教材。

本书承蒙西安交通大学陆丽娜教授审稿,对本书内容结构、编写大纲等方面提出了十分宝贵的意见;另外,西安交通大学计算机系统结构与网络研究所的唐亚哲、陈妍、朱海萍等许多同志对本书的编写也给予了关心和支持,在此一并表示衷心的感谢。

由于作者水平有限,加之 Java 语言的发展和变化非常迅速,书中难免有缺点、错误存在,欢迎同行专家和读者批评指正。

作者的电子邮件地址为:wangzhiwen51@sina.com。

王志文
于西安交通大学

目 录

第 1 章 Java 概述	(1)
1.1 Java 的诞生	(1)
1.2 Java 的技术特点	(2)
1.2.1 简明的语法结构	(2)
1.2.2 平台独立性	(3)
1.2.3 面向对象特征	(3)
1.2.4 面向网络环境	(4)
1.2.5 动态性	(4)
1.2.6 安全性	(4)
1.2.7 稳定性	(5)
1.2.8 多线程	(5)
1.2.9 类库丰富	(5)
1.3 Java 与 C/C++ 的差异	(6)
1.4 Java 程序运行机制	(7)
1.5 Java 运行时刻环境	(7)
1.5.1 Java 虚拟机	(8)
1.5.2 Java 平台	(9)
1.5.3 字节码介绍	(9)
1.5.4 编译与执行过程	(10)
1.5.5 垃圾收集	(10)
1.5.6 安全性问题	(11)
1.6 Java2 SDK	(12)
1.7 小结	(13)
习题	(13)
 第 2 章 Java 开发环境和程序范例	(14)
2.1 Java 程序开发环境介绍	(14)
2.1.1 JDK 软件包及其配置	(14)
2.1.2 JBuilder 开发平台	(17)
2.2 HelloWorld 应用程序	(18)
2.2.1 编辑源程序	(18)
2.2.2 应用程序结构剖析	(19)

2.2.3 编译并运行 HelloWorld 应用程序	(20)
2.3 编写 Applet 程序	(20)
2.3.1 Applet 的定义	(20)
2.3.2 第一个 Applet 程序: HelloWorld	(21)
2.3.3 Applet 程序结构剖析	(21)
2.3.4 运行 Applet 程序	(22)
2.4 小结	(24)
习题	(24)
第 3 章 Java 程序设计基础	(25)
3.1 Unicode 符号集	(25)
3.1.1 标识符	(25)
3.1.2 关键字	(26)
3.1.3 常量	(26)
3.1.4 运算符	(26)
3.1.5 分隔符	(26)
3.2 变量	(26)
3.2.1 基本数据类型	(26)
3.2.2 数据类型转换	(27)
3.2.3 变量定义和声明	(28)
3.2.4 变量命名	(30)
3.2.5 变量初始化	(30)
3.2.6 变量作用域	(31)
3.3 常量	(32)
3.3.1 整型常量	(32)
3.3.2 浮点型常量	(32)
3.3.3 布尔常量	(33)
3.3.4 字符常量	(33)
3.3.5 字符串常量	(33)
3.4 表达式与运算符	(34)
3.4.1 表达式	(34)
3.4.2 表达式的自动类型提升	(34)
3.4.3 运算符	(35)
3.4.4 运算符优先级	(35)
3.4.5 算术运算符	(36)
3.4.6 关系运算符	(38)
3.4.7 布尔逻辑运算符	(38)
3.4.8 位运算符	(39)
3.4.9 条件运算符	(40)

3.4.10 赋值运算符	(40)
3.5 对象与字符串运算符	(41)
3.5.1 对象运算符	(41)
3.5.2 字符串运算符	(42)
3.5.3 字符运算	(44)
3.6 数组	(44)
3.6.1 定义数组	(44)
3.6.2 创建数组	(45)
3.6.3 初始化数组	(45)
3.6.4 访问数组	(46)
3.6.5 多维数组	(46)
3.7 控制语句	(48)
3.7.1 条件语句	(48)
3.7.2 循环语句	(51)
3.7.3 转移语句	(54)
3.8 小结	(57)
习题	(57)
 第4章 Java 与面向对象技术	(59)
4.1 面向对象基础	(59)
4.1.1 面向过程编程与面向对象编程	(59)
4.1.2 面向对象编程的基本概念	(59)
4.1.3 面向对象编程的优点	(60)
4.2 类	(60)
4.2.1 类的结构	(60)
4.2.2 声明类	(61)
4.2.3 构造方法	(62)
4.2.4 析构方法	(62)
4.2.5 类修饰符	(62)
4.2.6 成员变量	(63)
4.2.7 特殊变量(null, this, super)	(65)
4.2.8 类转换	(66)
4.3 对象生命周期	(66)
4.3.1 创建对象	(66)
4.3.2 使用对象	(68)
4.3.3 释放对象	(68)
4.4 方法	(69)
4.4.1 方法定义	(69)
4.4.2 方法修饰符	(69)

4.4.3	方法重载	(70)
4.4.4	方法覆盖	(70)
4.4.5	main 方法	(71)
4.5	抽象类和接口	(71)
4.5.1	抽象类	(71)
4.5.2	接口	(72)
4.6	包	(73)
4.6.1	声明	(74)
4.6.2	加载	(74)
4.7	面向对象程序设计简单实例	(75)
4.8	小结	(77)
	习题	(77)
第 5 章	事件处理	(78)
5.1	事件	(78)
5.2	基于继承的事件模型	(78)
5.2.1	覆盖事件的处理方法	(78)
5.2.2	事件传递	(80)
5.2.3	构件标识	(84)
5.3	基于授权的事件模型	(85)
5.3.1	继承事件模型的不足	(85)
5.3.2	授权事件模型	(86)
5.3.3	事件类	(87)
5.3.4	事件监听者	(87)
5.3.5	适配器	(88)
5.3.6	基于授权事件模型的优点	(90)
5.4	高级事件处理	(91)
5.4.1	构件事件和语义事件	(91)
5.4.2	输入事件的消耗	(91)
5.4.3	语义事件	(92)
5.4.4	调度用户事件	(93)
5.5	小结	(95)
	习题	(95)
第 6 章	异常处理	(96)
6.1	Java 的程序错误处理机制	(96)
6.2	异常的分类	(97)
6.2.1	异常产生原因	(97)
6.2.2	异常分类	(97)

6.2.3 Throwable 类	(98)
6.3 异常的捕捉和处理	(98)
6.3.1 捕捉异常	(98)
6.3.2 异常的嵌套	(100)
6.3.3 finally 关键字	(101)
6.3.4 抛出异常	(102)
6.4 自定义异常类	(103)
6.5 异常处理的限制	(106)
6.6 小结	(107)
习题	(107)

第7章 输入和输出	(108)
7.1 流	(108)
7.2 java.io	(108)
7.3 输入流	(109)
7.3.1 InputStream	(109)
7.3.2 FileInputStream	(110)
7.3.3 ByteArrayInputStream	(110)
7.3.4 StringBufferInputStream	(111)
7.3.5 SequenceInputStream	(111)
7.3.6 PipedInputStream	(111)
7.3.7 FilterInputStream	(112)
7.4 输出流	(113)
7.4.1 OutputStream	(113)
7.4.2 FileOutputStream	(113)
7.4.3 ByteArrayOutputStream	(113)
7.4.4 PipedOutputStream	(114)
7.4.5 FilterOutputStream	(114)
7.5 UTF 字符流	(114)
7.5.1 UTF	(114)
7.5.2 字符流 Reader 和 Writer	(115)
7.6 高级流	(116)
7.7 文件	(117)
7.7.1 File 类	(117)
7.7.2 RandomAccessFile 类	(120)
7.8 流的应用范例	(121)
7.8.1 StringBufferInputStream	(121)
7.8.2 文件输入/输出流	(122)
7.8.3 管道流与线程通信	(123)

7.8.4 存储器读/写	(125)
7.9 小结	(127)
习题	(127)
第8章 AWT与布局管理器	(128)
8.1 AWT	(128)
8.1.1 AWT 简介	(128)
8.1.2 AWT 类库层次	(129)
8.2 基本构件	(129)
8.2.1 基本构件类层次结构	(129)
8.2.2 标签	(130)
8.2.3 按钮	(130)
8.2.4 复选框	(131)
8.2.5 单行文本框	(131)
8.2.6 多行文本框	(131)
8.2.7 列表框	(132)
8.3 容器构件	(132)
8.3.1 容器构件特征	(132)
8.3.2 容器构件类层次结构	(133)
8.3.3 画布	(133)
8.3.4 面板	(133)
8.3.5 窗口	(133)
8.3.6 框架	(137)
8.3.7 对话框	(138)
8.3.8 文件对话框	(138)
8.4 菜单	(139)
8.4.1 菜单基本结构及其创建方法	(139)
8.4.2 菜单使用范例	(140)
8.5 布局管理器	(142)
8.5.1 布局管理器的功能和特点	(142)
8.5.2 构件的首选尺寸	(143)
8.5.3 强制容器布置构件	(143)
8.6 标准布局管理器	(146)
8.6.1 FlowLayout 布局管理器	(146)
8.6.2 BorderLayout 布局管理器	(146)
8.6.3 CardLayout 布局管理器	(147)
8.6.4 GridLayout 布局管理器	(148)
8.7 GridBagLayout 布局管理器	(148)
8.7.1 约束变量	(148)

8.7.2 应用范例	(150)
8.8 null 布局管理器	(153)
8.9 小结	(153)
习题	(153)
第 9 章 Swing 构件	(155)
9.1 Swing 构件特征	(155)
9.2 轻量构件和重量构件	(156)
9.3 Swing 构件体系层次结构	(157)
9.4 Jcomponent 构件类	(157)
9.5 Swing 构件	(159)
9.6 Swing 构件应用	(159)
9.7 小结	(165)
习题	(165)
第 10 章 Applet 程序	(166)
10.1 Applet 模型	(166)
10.1.1 Applet 与 Web	(166)
10.1.2 Applet 运行时刻环境	(167)
10.1.3 Applet 的限制	(168)
10.1.4 Applet 的安全下载	(168)
10.2 Applet 程序	(169)
10.2.1 什么是 Applet 程序	(169)
10.2.2 Applet 程序的生命周期	(169)
10.2.3 日期显示小应用程序	(173)
10.3 获取资源	(174)
10.3.1 Image	(176)
10.3.2 AudioClip	(176)
10.3.3 URL	(176)
10.4 获取参数	(177)
10.5 小结	(180)
习题	(180)
第 11 章 多线程	(181)
11.1 多线程概念	(181)
11.2 Java 多线程技术	(182)
11.3 线程的建立	(183)
11.3.1 扩展 Thread 类	(183)
11.3.2 Runnable 接口	(185)

11.3.3	创建线程	(186)
11.3.4	线程的优先级	(189)
11.4	线程控制和管理	(191)
11.4.1	线程的生命周期	(191)
11.4.2	线程调度	(193)
11.4.3	线程通信	(196)
11.4.4	线程同步	(198)
11.4.5	死锁	(202)
11.5	线程分组	(202)
11.6	精灵线程与用户线程	(203)
11.7	多线程应用实例	(203)
11.8	小结	(206)
	习题	(207)
第 12 章	网络编程	(208)
12.1	Socket 通信	(208)
12.1.1	Socket 类	(208)
12.1.2	ServerSocket 类	(209)
12.1.3	InetAddress 类	(210)
12.1.4	Socket 通信过程和传输属性	(211)
12.1.5	创建 Socket	(212)
12.1.6	关闭 Socket	(213)
12.2	Socket 通信程序范例	(213)
12.3	支持多客户连接的 Socket 通信	(218)
12.4	DatagramSocket 通信	(221)
12.4.1	两个基本类	(221)
12.4.2	服务器程序	(222)
12.4.3	客户程序	(224)
12.5	多播传输	(224)
12.5.1	MulticastSocket 类	(225)
12.5.2	MulticastSocket 类的应用	(225)
12.6	URL	(227)
12.7	Java 的互联网协议	(228)
12.8	小结	(230)
	习题	(230)
第 13 章	Java 安全性	(231)
13.1	引言	(231)
13.2	类装载器	(231)

13.2.1 装载类文件·····	(231)
13.2.2 自定义类装载器·····	(232)
13.3 字节码验证·····	(233)
13.4 安全管理器和权限·····	(233)
13.5 Java2 平台安全机制 ·····	(234)
13.5.1 安全策略模型·····	(234)
13.5.2 权限使用·····	(235)
13.6 安全策略文件·····	(236)
13.6.1 工作机制·····	(236)
13.6.2 文件格式·····	(236)
13.6.3 策略文件应用·····	(238)
13.6.4 策略文件范例·····	(238)
13.7 小结·····	(239)
习题·····	(240)
第 14 章 JDBC 编程 ·····	(241)
14.1 JDBC 概述 ·····	(241)
14.1.1 ODBC 技术 ·····	(241)
14.1.2 JDBC 技术 ·····	(242)
14.1.3 JDBC 构成 ·····	(243)
14.1.4 JDBC 使用方法 ·····	(244)
14.2 JDBC 基本编程概念 ·····	(245)
14.2.1 JDBC URL ·····	(245)
14.2.2 加载驱动程序·····	(246)
14.2.3 创建数据库连接·····	(246)
14.2.4 DriverManager 类 ·····	(247)
14.2.5 创建 SQL 语句对象 ·····	(248)
14.2.6 Statement 接口 ·····	(248)
14.2.7 PreparedStatement 接口 ·····	(249)
14.2.8 ResultSet 接口 ·····	(250)
14.3 JDBC 应用范例 ·····	(252)
14.3.1 创建新的 ODBC 数据资源 ·····	(252)
14.3.2 JDBC 编程的基本步骤 ·····	(253)
14.3.3 范例程序代码·····	(254)
14.4 小结·····	(257)
习题·····	(257)
第 15 章 RMI 编程技术 ·····	(258)
15.1 RMI 编程概述 ·····	(258)

15.1.1	RMI 编程思想	(258)
15.1.2	RMI 分布式对象应用程序的核心功能	(259)
15.1.3	RMI 体系结构	(260)
15.2	RMI 类和接口	(261)
15.2.1	java.rmi.Remote 接口	(261)
15.2.2	java.rmi.RemoteException 类	(261)
15.2.3	java.rmi.server.RemoteObject 类及其子类	(262)
15.2.4	java.rmi.registry.LocateRegistry 类	(262)
15.2.5	java.rmi.Naming 类	(263)
15.2.6	java.rmi.server.RemoteServer 类	(264)
15.2.7	java.rmi.server.UnicastRemoteObject 类	(264)
15.2.8	java.rmi.RMISecurityManager 类	(265)
15.3	RMI 编译器(rmic)	(265)
15.4	RMI 编程示例	(266)
15.4.1	定义和实现远程接口	(266)
15.4.2	编写 RMI 服务器程序	(267)
15.4.3	编写 RMI 客户端程序	(268)
15.4.4	安全策略文件	(269)
15.4.5	批处理文件	(270)
15.5	程序运行结果	(271)
15.6	在不同机器上运行 RMI 程序	(272)
15.6.1	rmiregistry	(272)
15.6.2	程序修改	(272)
15.6.3	运行	(273)
15.7	小结	(273)
	习题	(273)
第 16 章	Java 与 CORBA	(274)
16.1	CORBA 简介	(274)
16.2	CORBA 体系结构	(274)
16.3	CORBA 工作原理	(275)
16.3.1	ORB	(276)
16.3.2	IOR	(276)
16.3.3	CORBA 协议栈	(277)
16.3.4	服务请求实现方式	(278)
16.4	Java 与 CORBA 的互补性	(279)
16.5	使用 Java 开发简单的 CORBA 应用	(279)
16.5.1	定义 IDL 接口	(279)
16.5.2	使用 idltojava 转换接口文件	(280)

16.5.3	idltojava 生成的 Java 文件	(280)
16.5.4	CORBA 服务器	(281)
16.5.5	CORBA 客户机	(283)
16.5.6	范例程序运行结果	(285)
16.5.7	在不同计算机上运行范例程序	(286)
16.6	小结	(286)
	习题	(286)
第 17 章	JNI 技术	(287)
17.1	概述	(287)
17.1.1	JNI 定义	(287)
17.1.2	使用 JNI	(287)
17.2	JNI 编程过程	(288)
17.3	小结	(292)
	习题	(292)
第 18 章	应用编程实例	(293)
18.1	Web 服务器	(293)
18.1.1	HTTP 协议	(293)
18.1.2	Web 服务器程序代码	(294)
18.1.3	程序结构分析	(298)
18.1.4	显示 Web 页面	(300)
18.1.5	运行实例	(301)
18.2	分布式数据库操作	(301)
18.2.1	定义远程接口	(302)
18.2.2	实现远程接口	(302)
18.2.3	服务器程序	(305)
18.2.4	客户程序	(306)
18.2.5	安全策略文件	(308)
18.2.6	运行程序	(308)
18.3	声音播放	(309)
18.3.1	声音文件类型	(309)
18.3.2	Applet 播放音频	(310)
18.3.3	Appplcation 播放音频	(312)
18.4	小结	(313)
	习题	(313)
附录 1	Java 语言参考	(314)
附录 2	Java 语言编程规范	(318)

Java 编程语言的诞生对于后来的网络应用产生了无比深远的影响,它的一系列技术特点使得它迅速流行并成为目前应用领域最为广泛的编程语言,尽管 Java 语言与传统的 C/C++ 有些类似,但它们之间还是存在相当程度的区别。作为介绍 Java 编程语言的开始,本书将介绍 Java 程序运行机制以及它的运行时刻环境,这些内容对于理解 Java 程序的跨平台特性有着非常重要的意义。

Java 语言诞生于 1991 年,是由 Sun 公司成功开发的新一代编程语言,它的最大特点在于使用它能够在各种不同种类机器、不同种类操作平台的网络环境下进行软件的开发。Java 是一种跨平台语言,适用于分布式计算环境的面向对象的编程语言,它具有可移植性、高度的安全性、简单性、与体系结构无关性以及动态执行等一系列优良特征。近年来,随着 Internet 的兴起和分布式计算环境框架技术的成熟,Java 正在逐步成为 Internet 应用的主要开发语言并成为全球信息网络舞台上一颗璀璨夺目的明星,它彻底地改变了应用软件的开发模式,在计算机领域里掀起了又一次技术革命高潮,为迅速发展的信息世界增添了新的活力。尤其是 Java2 平台的推出,进一步促进和加快了 Java 技术的应用。

需要声明的是:本书内容都是基于 Java2 平台而言的。

1.1 Java 的诞生

1991 年,Sun 公司为了开拓消费类电子产品市场(例如交互式电视、互动影像等),成立了一个专门独立开发小组,命名为“Green”。该小组的领导人是 James Gosling,他是一位非常杰出的程序员。在研究开发中,Gosling 深刻体会到了消费类电子产品的特点,它们要求可靠性高、费用低、标准化、使用简单。为了使整个系统脱离具体平台的依赖——平台无关性,Gosling 决定开始着手改写 C 编译器,但是他很快就发现仅仅依靠 C 是难以满足需求的,于是,他决定重新设计并开发出一种新的编程语言。

Green 项目小组开发的最初产品包括四个组成部分:Oak,GreenOs(一种专门的操作系统),用户接口模块以及硬件模块,这四个部分被集成到一个名为“*7”的类似 PDA 的电子设备中。成功地演示了这个系统以后,1993 年,Sun 公司成立了一个名为“FirstPerson”的子公司,不幸的是,这个系统在随后的几次大型商业活动中都惨遭失败。用 Java 创始人 James Gosling 的话说,他们发现“消费类电子工程市场并不是真实的,他们所期盼的远远超过了目前所能做到的”。尽管如此,Green 项目小组坚持他们的使命,并得到了来自大公司的源源不断的支持,一群才干横溢的计算机专家可以集中精力研究和解决在网络应用(networking

application)体系框架构造过程中出现的各种问题,最终,他们在技术上获得了成功。

1994年,WWW已经如火如荼地发展起来了。Gosling此时意识到WWW迫切需要一个中性的浏览器,它不依赖于任何硬件平台和软件平台,而且它还必须是一个实时性较高、可靠安全、有交互功能的浏览器。同时,尽管WWW页面的内容丰富,但是由于它是静态的,因此需要一种机制来使它具有动态性,增强动感。嵌入式语言支持是一种解决方案,但是这种语言必须很简洁,此外,安全性对于WWW服务而言也是一个十分令人头疼的问题,对于这些题,Oak具有先天的巨大优势。于是,Gosling决定首先将Java与WWW相结合。

Gosling决定使用Java开发一个新的Web浏览器。到1994年秋天,完成了WebRunner的开发工作。WebRunner是HotJava的前身,这个原型系统展示了Java可能带来的广阔市场前景。1995年1月,Oak被正式更名为Java(由于早先已经存在一种称为Oak的计算机语言,它被设计用来在应用框架内部工作,为了避免商标侵权行为而更改名字),它于1995年5月23日发表后,在业界引起了巨大的轰动。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语言被美国的著名杂志PC Magazine评为1995年10大优秀科技产品,随之而来的是出现了大量用Java编写的软件产品,并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”。

有人预言:Java将是网络上的“世界语”,今后所有用其他语言编写的软件通通都要用Java语言来改写。

1.2 Java的技术特点

Java不仅仅只是一个广泛使用的网络编程语言,它还代表了一种全新的计算概念和程序设计思想,它的主要技术特点表现在以下几个方面:

1.2.1 简明的语法结构

Java采用了与C相类似的语法结构。如果是一个熟悉C和C++的程序员就会发现,Java的语法结构几乎和C++完全相同(实质上还是存在少量差别),而业界公认C语言是一种效率很高、灵活性极强的计算机语言,这也是使用C语言开发软件的程序员数量十分庞大的根本原因。因此,采用类似C的语法结构,程序员能够很容易地将C及C++程序和Java程序进行相互转换。

Java语言的发明者拥有丰富的C和C++语言的使用经验,他们在吸收C和C++优秀特征的同时,舍弃了其中一些容易引起问题,且不是绝对必要的部分功能。

指针是C和C++的重要特征之一,甚至有人说过,指针是C的灵魂。指针是C和C++语言完成一些高级程序设计的基础,使用指针,可以灵活高效地解决很多底层问题。然而,指针也不是完美无缺的,指针好比是一把双刃剑,一旦使用失误,就有可能给用户带来灾难性的后果。Java语言决定取消指针,而仅仅保留了引用(reference)。这样,通过引用的使用,不但保留了指针的灵活性,而且避免了许多由于指针的存在而潜在的隐患。

另外,Java语言的使用也比较简单。由于Java语言是一种面向对象的语言,它通过提供最基本的方法来完成指定的任务,只需理解一些基本的概念,就可以用它编写出适合于各种情