



高等院校计算机应用技术规划教材

Java 程序设计实用教程

秦学礼 主编 孙伟 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高等院校计算机应用技术规划教材

Java 程序设计实用教程

秦学礼 主 编

孙 伟 副主编

郑 颖 陈 薇 李向东 杨 昕 编 著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书从 Java 语言最基本的概念开始讲述,详细介绍了 Java 语言的开发工具 J2SDK 和 JBuilder 9.0 集成开发环境的使用,使读者能够在学习的初期就熟练地应用开发工具进行试验和验证各种技术细节,能比较快速地进入学习环境,也为进一步开发应用程序打下良好的基础。

本教材还对 Java 面向对象程序设计的基本概念和技术等内容进行了较为详细的讲解,通过大量的编程实例讲述使用 Java 语言及其类库编程,对 Java 的输入/输出处理、图形界面的设计等也进行了深入的介绍。另外本书每章都安排了大量有针对性的习题,还给出了一些综合应用的实例。

本书可作为高等院校、高职高专和 Java 技术培训的教材和教学参考书,也可作为 Java 编程爱好者的入门参考书。

图书在版编目(CIP)数据

Java 程序设计实用教程/秦学礼主编. —北京:中国铁道出版社,2006.1
高等院校计算机应用技术规划教材
ISBN 7-113-06831-6

I. J... II. 秦... 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(2006)第 005228 号

书 名: Java 程序设计实用教程

作 者: 秦学礼 孙 伟 等

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟 秦绪好

责任编辑: 严 力 谢立和 翟玉峰

封面设计: 薛 为

封面制作: 白 雪

印 刷: 北京市兴顺印刷厂

开 本: 787×1092 1/16 印张: 20.25 字数: 489 千

版 本: 2006 年 2 月第 1 版 2006 年 2 月第 1 次印刷

印 数: 1~5 000 册

书 号: ISBN 7-113-06831-6/TP·1687

定 价: 26.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

计算机网络正在以前所未有的速度迅猛发展,在网络程序设计应用领域,从电子商务、远程教学到网络游戏等都在纷纷使用 Java 技术,Java 手机编程和基于 Java 技术的各种芯片的应用等在日常生活中也随处可见。因此,Java 语言的编程技术正逐步成为计算机网络程序设计开发的主流。

Java 语言不仅可以用来开发大型的应用程序,而且在 Internet 上有着重要而广泛的应用。Java 确实具备了“一次撰写,到处运行”的特点,尤其是 Java Swing 推出之后,使 Java 的功能更加强大。

本书是一本实用教程,通俗易懂,实例丰富。针对难理解的问题,例子都是从简单到复杂,逐步深入,便于读者掌握 Java 编程技巧。本书主要包括 Java 语言基础、Java 面向对象编程基础、图形用户界面与事件处理、数组、向量及字符处理、窗口、菜单和对话框、Java Swing 基础、多线程、输入/输出处理、Java 数据库连接(JDBC)等内容。

本书不仅可以作为高等院校、成人高等院校和各种 Java 语言培训班的教材,也适合没有任何编程经验的初学者使用。要求熟练掌握 Java 程序设计的基本知识、基本概念;掌握面向对象程序设计的基本思想和 Java 程序设计的编程方法和技能;掌握可视化创建用户界面的方法和技能及运用 JDBC 的数据库编程的能力。

本书内容新颖、结构合理、概念清晰、通俗易懂、实用性强。例题的选择既考虑加深对知识的理解和掌握,又考虑到学生的学习兴趣和编程应用,并对例题进行了详细的讲解和分析。每章习题的编写具有较强的针对性,以帮助学生巩固所学知识和提高实际编程能力。根据学生基础不同和讲述内容的取舍不同,本书的教学可安排 60~70 课时,其中上机试验应在 50 学时以上。

特别是在第 1 章和第 2 章的学习过程中,对 Java 程序的开发和运行环境、Java 语言的基础要有基本的掌握,尽可能使学生通过本章的学习,对 Java 语言及其编程环境建立起基本的概念。为了便于后面章节内容的学习,这两章的内容还需要学生在学习过程中多次复习和查阅。在上机练习过程中要熟练地使用开发工具并掌握程序调试的方法,为后面的学习和进一步开发打下基础。

本教材的主要特点:

(1) 针对程序设计教学的规律,精选课程内容,对过难或偏离程序本质的内容进行精简,以提高程序设计能力为主线贯穿本教程的始终。介绍程序设计算法和实例讲解,以便学习和理解基本原理、方法和技术。

(2) Java 语言是一门实践性很强的课程,本书贯彻从实践中来到实践中去的原则,课堂教学与上机并重。部分章节习题中包含编程题。

(3) 实例完整、丰富多彩、实用面广。突出可视化用户界面的编程技能的培养,注重数据库应用程序设计的编程能力的提高。

(4) 参加本教材编写的教师都是教学第一线的教师,经过多年教授“Java 程序设计”课程的实践,积累了一定的教学经验和体会。针对初学者的特点,在内容取舍、顺序编排、

概念描述和讲述方法等方面都做了较多的探讨和实践，力争能较好地适合初学者的使用。

(5) 讲述内容紧跟 Java 语言及其相关技术的发展。

本书第 1、2 章由秦学礼编写，第 3、4 章由郑颖编写，第 5、6 章由陈薇编写，第 7、8 章由李向东编写，第 9、10 章由杨昕编写，第 11、12 章由孙伟编写。全书由秦学礼策划、组织编写、修改校对和统稿。

本书提供的全部程序在 J2SDK 和 JBuilder 9.0 集成开发环境中调试通过，源程序可以与铁道出版社计算机图书中心教材部联系索取。

由于作者水平有限，书中可能会有错误或不妥之处，欢迎批评指正。联系邮件：QINXUELI@ZJ.COM 或 XUELI_QIN@HOTMAIL.COM。

在本书顺利出版之际，感谢林美雄教授认真审读了全文并提出了许多修改意见。本书还参考了许多作者的书籍和资料，在这一并表示深深的谢意。

秦学礼

2005 年 11 月于杭州编

第 1 章 Java 语言概述	1
1.1 Java 语言介绍	1
1.1.1 Java 语言出现的背景、影响及应用前景	1
1.1.2 Java 语言的特点	2
1.1.3 Java 语言的高级应用	4
1.2 Java 程序开发过程	6
1.2.1 编写 Java 程序源代码	6
1.2.2 编译 Java 源程序生成字节码	6
1.2.3 Java 程序的运行	7
1.3 Java 开发环境	8
1.3.1 开发和运行 Java 的硬件环境	8
1.3.2 开发和运行 Java 的软件环境	8
1.3.3 JDK 开发环境	8
1.3.4 JDK 的安装与环境设置	9
1.3.5 JBuilder 集成开发环境	11
1.3.6 JBuilder 集成开发环境的安装要求	12
1.3.7 JBuilder 集成开发环境的安装	13
1.3.8 JBuilder 9.0 集成开发环境的界面	14
1.4 程序实例	16
1.4.1 简单的 Java Application 应用程序	16
1.4.2 简单的 Java Applet 小应用程序	17
1.4.3 用 JBuilder 9 集成开发环境创建一个工程	19
习题 1	21
第 2 章 Java 语言基础	23
2.1 Java 源程序与书写规范	23
2.1.1 Java 源程序结构	23
2.1.2 Java 程序书写规范	23
2.2 Java 程序基本结构	24
2.2.1 保留字 (Reserved Word)	24
2.2.2 标识符 (Identifiers)	25
2.2.3 注释	25
2.2.4 分隔符	26
2.3 Java 数据类型	27
2.3.1 常量	28

2.3.2	变量	28
2.3.3	基本数据类型	30
2.3.4	字符串	33
2.4	Java 运算符和表达式	35
2.4.1	算术运算符	35
2.4.2	关系运算符	38
2.4.3	逻辑运算符	39
2.4.4	位运算符	40
2.4.5	字符串运算符	41
2.4.6	条件运算符	42
2.4.7	其他运算符	42
2.4.8	表达式	43
2.5	Java 控制语句	43
2.5.1	分支语句	44
2.5.2	循环语句	46
2.5.3	特殊的控制流语句	49
2.6	Java 类定义规范	51
2.6.1	方法的定义	51
2.6.2	方法的调用及参数传递	53
2.6.3	方法的返回值	56
2.6.4	递归方法	56
2.7	Java 开发类库组成	57
2.7.1	Java 类库结构	57
2.7.2	基础类库	58
2.7.3	常用基础类介绍	59
2.8	程序实例	62
2.8.1	实例 1	62
2.8.2	实例 2	63
习题 2		64
第 3 章	Java 面向对象编程基础	66
3.1	类	66
3.1.1	类的概念	66
3.1.2	类的定义	67
3.1.3	类的构造方法	68
3.1.4	类的封装、多态和继承	70
3.2	对象	75
3.2.1	对象的概念	75
3.2.2	创建对象	75

3.2.3 对象的引用	76
3.3 类的成员变量	76
3.3.1 成员变量定义的语法规范	76
3.3.2 成员变量修饰符	77
3.4 类的成员方法	79
3.4.1 方法定义的语法规范	79
3.4.2 成员方法的方法修饰符	80
3.4.3 使用 <code>this</code> 关键字修饰的方法	82
3.5 程序实例	82
3.5.1 实例 1	82
3.5.2 实例 2	84
习题 3	86
第 4 章 包、接口和异常	89
4.1 包和接口	89
4.1.1 包	89
4.1.2 接口	91
4.2 异常	93
4.2.1 异常与异常类	93
4.2.2 异常处理	95
4.2.3 创建用户自己的异常	98
4.3 程序实例	98
4.3.1 实例 1	98
4.3.2 实例 2	100
习题 4	102
第 5 章 数组、向量及字符处理	105
5.1 数组 (Array)	105
5.1.1 一维数组	105
5.1.2 二维数组	108
5.2 向量 (Vector)	113
5.2.1 创建向量类的对象	114
5.2.2 访问向量序列中的元素	114
5.2.3 在向量序列中添加元素	115
5.2.4 修改或删除向量序列中的元素	116
5.2.5 查找向量中的元素	117
5.2.6 测试向量中元素个数	118
5.3 字符处理 (String)	118
5.3.1 String 类	119
5.3.2 StringBuffer 类	122

习题 5	125
第 6 章 输入/输出处理	127
6.1 I/O 流概述	127
6.2 字节流	127
6.2.1 InputStream 类和 OutputStream 类	127
6.2.2 文件流 FileInputStream/FileOutputStream	128
6.2.3 标准流	131
6.3 字符流	132
6.3.1 Reader 类和 Writer 类	132
6.3.2 InputStreamReader 类和 OutputStreamWriter 类	134
6.3.3 BufferedReader 类和 BufferedWriter 类	135
习题 6	137
第 7 章 Applet 与图形用户界面	139
7.1 Applet	139
7.1.1 Applet 基本工作原理	139
7.1.2 Applet 的运行控制及生命周期	141
7.1.3 Applet 与 HTML 文件的配合	141
7.2 常用组件与事件处理	144
7.2.1 标签	146
7.2.2 按钮	147
7.2.3 文本框和文本区	149
7.2.4 下拉列表	153
7.2.5 复选框和选项按钮	154
7.2.6 列表框	156
7.3 组件布局管理	158
7.3.1 FlowLayout 布局	158
7.3.2 BorderLayout 布局	159
7.3.3 GridLayout 布局	159
7.3.4 CardLayout 布局	160
7.3.5 GridBagLayout 布局	162
7.3.6 null 布局	164
习题 7	165
第 8 章 窗口、菜单和对话框	166
8.1 窗口及事件	166
8.2 菜单	170
8.2.1 为窗口加入菜单和定义二级菜单	170
8.2.2 弹出式菜单	173
8.3 对话框	174

8.3.1	Dialog 类	174
8.3.2	FileDialog 类	177
习题 8		179
第 9 章	Java Swing 基础	181
9.1	重要的类	181
9.2	中间容器	182
9.3	各种组件	183
9.3.1	JFrame 类	183
9.3.2	标签 (JLabel)	184
9.3.3	按钮 (JButton)	184
9.3.4	单选按钮组和复选框	187
9.3.5	组合框 (JComboBox)	191
9.3.6	列表框 (JList)	193
9.3.7	文本框 (JTextField)	195
9.3.8	文本区 (JTextArea) 组件	198
9.3.9	进度条 (JProgressBar)	200
习题 9		202
第 10 章	多线程与网络编程	204
10.1	多线程	204
10.1.1	多线程基本概念	204
10.1.2	创建线程的方式	206
10.1.3	线程的控制	207
10.1.4	线程的优先级及终止	208
10.1.5	多线程的互斥与同步	209
10.2	网络编程	214
10.2.1	网络基本概念	214
10.2.2	Java 与 Internet	215
10.2.3	使用 InetAddress	215
10.2.4	使用 URL	219
10.2.5	Socket 通信	221
10.2.6	数据报通信	222
习题 10		223
第 11 章	Java 数据库连接 (JDBC)	224
11.1	JDBC 基础	224
11.1.1	JDBC 驱动程序	224
11.1.2	JDBC API 常用类和接口	225
11.2	配置 ODBC 数据源	228
11.3	JDBC 数据库编程	229

11.4 示例	232
11.5 程序实例	233
11.5.1 实例 1	233
11.5.2 实例 2	234
11.5.3 实例 3	236
11.5.4 实例 4	238
11.5.5 实例 5	239
习题 11	242
第 12 章 使用 JBuilder 9.0 开发应用程序	243
12.1 JBuilder 9.0 开发应用程序基础	243
12.1.1 JBuilder 工程	243
12.1.2 配置类库路径	247
12.1.3 开发 Java 应用程序	249
12.2 Java 应用程序的可视化设计	252
12.2.1 菜单的制作	252
12.2.2 界面设计	253
12.3 Java 应用程序开发实例	256
12.3.1 系统模块设计	256
12.3.2 数据库结构设计	257
12.3.3 详细设计	258
附录 A 练习与解答	273
习题 1	273
习题 2	274
习题 3	277
习题 4	278
习题 5	279
习题 6	281
习题 7	284
习题 8	287
习题 9	290
习题 10	294
习题 11	300
附录 B 下载文件的安装与使用	309
参考文献	311

1.1 Java 语言介绍

Java 语言不仅是一个程序设计语言，更是一个网络操作系统。因为：

- (1) Java 语言可不依赖于任何现有操作系统，而直接运行在某种机器或 CPU 上。
- (2) Java 语言有自己的内存管理机制。
- (3) Java 语言支持多线程程序，可对处理器资源进行分配和管理。
- (4) Java 语言可对设备进行管理，如屏幕、键盘、鼠标等。

(5) Java 语言支持网络文件管理，Java 语言可打开远程机器上的文件，Java 语言文件的概念已经扩充到整个 Internet 网络。

1.1.1 Java 语言出现的背景、影响及应用前景

1. Java 语言出现的背景

Java 语言诞生于 1991 年，它是由 Sun Microsystems 的一个开发小组在开发 Green 项目时完成的。该项目最初是开发一种用于消费类电子产品的与平台无关的软件技术，主要用来解决诸如电视机、电话、闹钟等家用电器的控制和通信问题。开始时，该项目小组成员准备采用 C++ 语言，但是他们感到 C++ 语言太复杂，安全性也差，无法满足项目设计的需要，最后决定基于 C++ 语言开发一种新的编程语言。计划的主持人 Jame Gosling 先生看到窗口的橡树，就将这种程序语言命名为 oak，可是发现另有程序语言已经命名为 oak，后来该计划的成员有一天去喝印尼爪哇岛出产的咖啡，因此就将该语言命名为 Java。当时并没有引起人们的注意，直到 1994 年下半年，随着 Internet 的迅猛发展，万维网 WWW 的快速增长，促进了 Java 语言研制的进展。SUN 公司在 1995 年的 5 月正式推出 Java，可用于制作动态网页、企业应用程序设计、全球信息网服务端程序设计等。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、Novell、Apple、DEC、SGI 等著名的计算机厂商。

2. Java 语言对软件开发技术的影响

Java 语言的出现，必然会引起一场软件革命，这是因为 Java 语言能在二进制上兼容，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。时至今日，Java 的优势已经不再局限于跨平台和动态页面显示，更多的优势逐步在支持 CORBA、Web 服务器后端处理以及应用整合等方面体现出来。

Java 语言对现在以及未来软件工程和软件发展可能产生的影响，可从如下几个方面考虑：

• 软件工程

Java 适用于 Web 的许多特征，如平滑移植、安全、健壮和可扩展等，为面向对象软件工程设计提供了实现工具和想像空间。遵循面向对象思想，软件生命周期的各个部分都将从面向对象的角度出发实现。比如用户需求分析可将用户的需求进行动态的、可视化的描述，以

提供设计者更加直观的要求。而用户的需求是各色各样的，不受地区、行业、部门、爱好的影响，都可以用 Java 语言描述清楚。

- 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 (Object Oriented) 的技术与方法来开发，这与 Java 的面向对象特性、模块化特性、安全性以及适应 IT 技术发展密不可分。这也是符合最新的软件开发规范要求的。

- Web 技术

Java 提供的 CGI、ASP 替代方法 servlet/jsp 以及 EJB 都是与互联网发展需求相呼应的，由此可见 Java 在未来必然紧跟互联网发展趋势，采用 Java 解决企业基于 Web 的应用是明智的选择。

- 图形处理与多媒体效果

Java 2 以及以后版本提供了 Swing、ZD、3D、Sound 等图形、图像和多媒体支持类库，弥补了早期 Java 版本在此方面的不足。Java 的媒体处理能力使程序开发在动画效果上远比 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态画面的资源。

- 分布式计算

目前 Java 产品主要有基于 Web 的实时信息发布系统、Web 邮件系统、用于企业分布式计算的消息传输系统、基于 servlet/jsp/ejb 开发的电子商务平台、提供 servlet/jsp/ejb 开发和运行环境的应用服务器、结合 CORBA (Common Object Request Broker Architecture) /XML 等技术开发的分布计算支撑平台等。另外用 Java 开发的多媒体软件可以实现集可视化、可操作化于一体的效果。

- 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，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。

3. Java 语言的应用前景

Java 语言有着广泛的应用前景，大体上可以从以下几个方面来考虑其应用：

- (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、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、控制流程等）。
- (5) Internet 的系统管理功能模块的设计，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 语言的特点

Java 语言具有能独立于软硬件平台而运行、面向对象、可对动态画面进行设计与操作、

稳定性高等特点,又具有多线程、内置校验器用来防止病毒入侵等功能,所以用在 Internet 上研制与开发软件时,特别受用户的欢迎。

Java 语言有以下特点:

1. 简单性

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况的应用程序。由于 Java 的结构类似于 C 和 C++,所以一般熟悉 C 与 C++语言的编程人员稍加学习就能掌握 Java 的编程技术。

Java 略去了运算符重载、多重继承等模糊的概念,并且实现了自动垃圾回收,Java 所具有的自动内存管理机制也大大简化了 Java 程序设计开发。另外,Java 也适合于在小型机上运行,它的基本解释器及支持类的模块只有 40KB 左右,加了标准类库和支持线程的模块也只有 215KB 左右。

2. 面向对象

简单地说,面向对象设计是一种以数据(对象)以及接口为重心的程序设计方法,面向对象的设计可以说是定义程序模块“即插即用”的机制。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。Java 的面向对象机制实际上可以看作是 C++面向对象机制的延伸。Java 提供了简单的类机制和动态的构架模型,对象中封装了它的状态变量和方法(函数、过程),实现了模块化和信息隐藏;而且提供了一类对象的原型,通过继承和重载机制,子类可以使用或者重新定义父类或者超类所提供的过程,从而实现代码的复用。

3. 自动内存管理

Java 的自动内存垃圾收集(Auto Garbage Collection)实现了内存的自动管理,因此简化了 Java 程序开发的工作,早期的 GC(Garbage Collection)对系统资源抢占太多而影响整个系统的运行,Java 2 对 GC 进行的改良使 Java 的效率有了很大提高。GC 的工作机制是周期性地自动回收无用存储单元。Java 的自动内存回收机制还在简化程序开发的同时,提高了程序的稳定性和可靠性。

4. 分布性

Java 是面向网络的语言。Java 还提供了专门针对互联网应用的类库,通过它提供的类库可以处理 TCP/IP 协议,用户可以在网络上很方便地访问其他对象,如 URL、Java mail 等。

5. 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。它提供自动垃圾收集来进行内存管理,防范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例外处理机制,在编译时,Java 揭示出可能出现但未被处理的例外,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。

6. 安全性

用于网络、分布环境下的 Java 必须要防止病毒的入侵。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,这样就防止了程序员使用“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。

7. 体系结构中立性

Java 解释器生成与体系结构无关的字节码指令,只要安装了 Java 运行时系统,Java 程序

就可在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示, Java 解释器得到字节码后, 对它进行转换, 使之能够在不同的平台运行。

8. 可移植性

与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同机器。同时, Java 的类库中也实现了与不同平台的接口, 使这些类库可以移植。另外, Java 编译器是由 Java 语言实现的, Java 运行时系统由标准 C 语言实现, 这使得 Java 系统本身也具有可移植性。

“Write once, run everywhere”也许是 Java 最诱人的特点。用 Java 开发的系统其移植工作几乎为零, 一般情况下只需对配置文件、批处理文件作相应修改即可实现平滑移植。

9. 跨异构环境

Java 是网络空间的“世界语”, 编译后的 Java 字节码是一种“结构中立性”(Architecture Neutral)的目标文件格式, 可以在所有提供 Java 虚拟机(JVM)的多种不同主机、不同处理器上运行。

10. 解释执行

Java 解释器(Interpreter)可以直接在任何已安装了解释器的机器上解释、执行 Java 字节代码, 不需重新编译。当然, 其版本向上兼容, 因此如果是高版本环境下编译的 Java 字节码到低版本环境下运行也许会有问题。

11. 高性能

和其他解释执行的语言如 BASIC、TCL 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, 从而得到较高的性能。

12. 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。

13. 动态性

Java 的设计使它适合应用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。

14. 异常处理

Java 采用异常模型使程序的主流逻辑变得更加清晰明了, 并且能够简化错误处理工作。

15. 可扩充

Java 目前发布的 J2EE 标准主要为采用 Java 技术的企业的全面解决方案提供了一个技术规范框架, 规划了一个利用现有和未来各种 Java 技术整合解决企业应用的远景蓝图。

1.1.3 Java 语言的高级应用

Java 技术根据硬件平台与适用环境的差异, 分成几个分支。JDK1.1 适用于一般消费性电子产品等。嵌入式系统的 Java 平台是 Personal Java 与 Embedded Java, 此二者并无明确的界线, 大致上来说, 运算资源、内存以及显示设备比较丰富的, 使用 Personal Java, 如 Set-Top Box (机顶盒)、可视电话等; 反之, 资源较有限的使用 Embedded Java, 如呼叫器、移动电话等。除了 PC 使用的 Java 平台、IA 使用的 Personal Java 与 Embedded Java 平台之外, Java Card

也是一个 Java 平台, 应用于 Smart Card (IC Card)。

Java 2 出现后, 推翻了先前的 Personal Java 与 Embedded Java 的分法, 改分成 Java 2 Platform Enterprise Edition (简称 J2EE)、Java 2 Platform Standard Edition (简称 J2SE)、Java 2 Platform Micro Edition (简称 J2ME)。J2EE 适用于 Web 应用服务器, 目前已经成为企业运算、电子商务等领域中相当热门的技术; J2SE 适用于一般的计算机; J2ME 适用于消费性电子产品。除了这三者之外, JavaCard 依然是独立的一套标准。

目前, Java 技术的架构包括 3 个方面:

(1) J2EE 企业版是以企业为环境, 开发应用程序的解决方案。

J2EE 已经成为开发商创建电子商务应用的事实标准。正是认识到 J2EE 平台作为一种可扩展的、全功能的平台, 可以将关键的企业应用扩展到任何 Web 浏览器上, 并适合多种不同的 Internet 数据流; 可连接到几乎任何一种传统数据库; 使企业经理根据多家企业所提供的产品和技术, 部署最佳的解决方案进而降低开发网络化应用的费用和复杂性这一巨大优势, 很多厂家都表示将对 J2EE 给予支持, 并将 J2EE 技术作为大型 B2B 市场和海量交易处理的安全稳定的端到端平台。J2EE 技术的基础就是 J2SE 标准版, 它巩固了标准版中的许多优点。其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的体系结构。它为灵活配置各种多层企业应用软件, 特别是 B2B、B2C 等电子商务应用, 提供了强大的服务功能。最近又新加了 Connector API 服务, 使企业应用的开发和部署有了一系列成熟的技术。

目前 J2EE 已经成为服务器运算环境的标准。Java Servlets、JSP (Java Server Pages)、EJB (Enterprise JavaBeans)、JavaMail、JDBC、JMS (Java Message Service) 等, 都是各家厂商产品开发的重点方向。J2EE 兼容的是一般 Intel 个人计算机 (Linux、Windows、...) 以及各种高效率、高稳定性的 UNIX 服务器主机, 未来必定成为服务器市场上的主要选择之一。

(2) J2SE 标准版为桌面开发和低端商务应用提供了可行的解决方案。

J2SE 是 Java 2 平台的标准版, 它适用于桌面系统, 提供 CORBA 标准的 ORB (Object Request Broker) 技术, 结合 Java 的 RMI 支持分布式交互操作环境。它运行在 Java 虚拟机上。在引入了 Java IDL (Interactive Data Language, 接口定义语言) 后, J2SE 支持 IIOP (互联网内部对象请求代理协议) 通信。它是高可移植性、异构性的实现环境和健壮平台, 也是实现可伸缩性、可移植性、分布式异构交互操作应用软件开发的标准平台。

在中型的 J2SE (Java 2 Standard Edition) 方面, Sun 推出一个新的解决方案, 称为 Java Web Start。原 Java Applet 是在 Web Browser 中间留出一块方形区域来执行 Java 程序, 但是这样在执行效率和兼容性上都受限于原有的 Web Browser。现在新推出的 Java Web Start 则是在操作系统上直接执行的 Java Application, 但是可以在网页上激活。

这样不仅可与页面结合, 在执行上也更快、更有效率。并且, Sun 和 IBM 都将推出支持 64 位运算的 Java 版本, 这对一般计算机上执行的客户端 Java 应用系统的开发将会是一大利器。

(3) J2ME 小型版是致力于消费产品和嵌入式设备的最佳解决方案。

J2ME 提供了 HTTP 高级 Internet 协议, 使移动电话能以 Client/Server 方式直接访问 Internet 的全部信息, 不同的 Client 访问不同的文件, 此外还能访问本地存储区, 提供最高效率的无线交流。J2ME 是 Java 2 平台的微型版, 它分成 CDC (Connected Device Configuration) 和 CLDC (Connected Limited Device Configuration) 两部分。CDC 运行在连接虚拟机上, 为笔记本电脑一

类较复杂的移动设备提供应用平台; CLDC 运行在核心虚拟机 (KVM) 上, 它实现 MIDP (Mobile Information Device Profile) 移动信息设备应用平台, 即针对手机之类的设备建立移动计算平台。

在小型的 J2ME (Java 2 Micro Edition) 方面, 主要是应用在内存容量小、体积也较小的电子装置上。小至智能卡、移动电话与个人数字助理都是运用 J2ME 的最佳平台。Java 在 Palm 的应用上, PalmOS 4.0 内含 KJava, Sun 也推出针对 PalmOS 使用的 J2ME 版本。所以, 以既有的 Java 程序设计知识, 就可以在 Palm PDA 上开发出 Palm 的各种应用系统。Java 和 Palm 这两个标准平台的结合, 将是未来 PDA 应用的趋势。在手机的应用上, Nokia、Motorola、Ericsson 都将推出利用 J2ME 技术的新手机, 所以 Java 程序设计师有更多的平台可供施展。此种结合 J2ME 及无线通信技术的无线开放应用平台, 将为移动商务提供极佳的解决方案。

除了以上这三大 Java 组合之外, Java 和 XML 的整合也是未来的发展方向。Sun 公司已经推出 Java 处理 XML 的标准延伸 API——Java API for XML Parsing (JAXP), 可以让各家所制作的 XML 解析器有接口上的标准。所以在 Java 程序中, 只要了解一套 API (JAXP) 就可以完全处理 XML 文件, 让 XML 的应用更加方便。Java 这个跨平台的开发环境, 加上 XML 这个跨平台的数据格式, 此种跨平台优势组合必将成为未来信息传递及数据交换的主要应用技术, 成为一个最佳的跨平台解决方案。

1.2 Java 程序开发过程

根据应用类型和运行环境的不同, Java 程序的开发可以分为 Java Application 和 Java Applet 两类。简单地讲, Java Application 是完整的程序, 需要独立的编译器和解释器来解释运行; 而 Java Applet 则是嵌在 HTML 编写的 Web 页面中的非独立程序, 由 Web 浏览器内置的 Java 解释器来解释执行。Java Application 和 Java Applet 各自使用的场合也不相同。

Java 程序的开发通常需要编写源代码、编译生成字节码和运行几个步骤。

1.2.1 编写 Java 程序源代码

Java 源程序是以 java 为后缀的文本文件, 可以用各种 Java 集成开发环境中源代码编辑器来编辑, 也可以用其他文本编辑工具, 如 Windows 系统的记事本、Word (要保存成 Java 文件) 等编辑软件, 编程者利用文本编辑器将源代码输入到计算机中, 形成一个文件保存起来。

在编写 Java 源程序的时候, 需要注意下面几个问题:

(1) Java 是区分大小写的语言, 关键字的大小写不能搞错, 如果把类 class 写成 Class 或者 CLASS, 都是错误的。

(2) 在一个类的内部不能定义其他的类, 即类和类之间是平行的, 而非嵌套的关系。

(3) 一个程序中可以有多个类, 但是只能有一个主类。不同类型的 Java 程序, 其主类的标志是不同的。

(4) 源程序编写好之后, 应该以文件的形式保存在硬盘或 U 盘上, 称为源程序文件或源文件。这个源文件的名称不是随便取的, 它必须与程序的主类名一致, 并且以“java”为后缀。

1.2.2 编译 Java 源程序生成字节码

高级语言程序从源代码到目标代码的生成过程称为编译。在 Java 程序中, 源代码经编译