

Java 2

技术内幕



(美) Stephen Potts 等著

马朝晖 等译



机械工业出版社
China Machine Press

Sun公司核心技术丛书

Java 2 技术内幕

(美) Stephen Potts 等著

马朝晖 等译



机械工业出版社
China Machine Press

本书是讲述Java技术的权威著作,全面介绍了Java 2 SDK 1.4的各种功能,内容涉及Java 2高级编程主题的方方面面,如分布式对象编程、企业级Java应用、Java集成技术、Java媒体技术、设计Java类的方式等。全书贯穿了一个用Java编写的实际应用,可供读者理解Java的各种概念和编程方法,并且所有源代码可以从网上下载。本书适合有一定Java语言基础的编程人员使用。

Authorized translation from the English language edition entitled Java 2 Unleashed by Stephen Potts, et al, published by Pearson Education, Inc, publishing as SAMS. Copyright © 2002 by SAMS.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by China Machine Press.

Copyright © 2003 by China Machine Press.

本书中文简体字版由美国Pearson Education培生教育出版集团授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2003-1016

图书在版编目(CIP)数据

Java 2技术内幕/(美)波茨(Potts, S.)等著;马朝晖等译.-北京:机械工业出版社, 2003.3

(Sun公司核心技术丛书)

书名原文: Java 2 Unleashed

ISBN 7-111-11621-6

I. J… II. ①波… ②马… III. Java语言-程序设计 IV. TP312

中国版本图书馆CIP数据核心字(2003)第009754号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:路晓村 周 睿

北京中加印刷有限公司印刷·新华书店北京发行所发行

2003年4月第1版第1次印刷

787mm×1092mm 1/16·49.25印张

印数: 0 001-4 000册

定价: 79.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

前 言

我们写作本书的目的是讲述Java程序员真正需要掌握的技能。程序员很少碰到只涉及Java语言本身的工程。现代工程往往涉及对核心语言的扩展，比如JDBC、RMI、J2EE、JavaBean和串行端口通信。程序员通常要用不同的扩展开发不同的工程。

我们先浏览了书店，发现有大量的入门书籍和许多针对专门主题的高深书籍，缺少中等程度的书，也就是用朴实无华的语言解释高级主题的书，这就是我们要提供的。本书有以下特点：

- 本书是由程序员用朴实无华的语言写成的。我们对读者的惟一要求是了解Java。例如，讨论Java Sound的一章假设你了解Java，但是不要求你了解Java Sound API。这样就很容易展开讨论。
- 每章包含你可以使用的代码。本书中的所有代码都可以复制进你的工程。本书很少使用无法运行的“代码片段”来进行解释。你可以复制或输入本书中的例子、运行它们并可修改它们以适合你的特定需要。
- 本书给出了每个例子的输出。如果例子的结果很难显示，我们会添加println（）调用，这样你就能够确定例子是否成功运行了。
- 你将节省时间。你可以参加一个课程或者在Internet上冲浪数小时寻找符合需要的资料，但是本书能够更快地提供信息。
- 本书比一套单一主题的书便宜得多。一本只讨论本书主题之一的书价格上往往就和本书相当了。收集20或30本这样的书可需要不少钱。在学习了本书中的一章之后，你可能想买一本关于此主题的专著。但是，你很可能发现你学到的东西已经足以帮助你理解在Internet上找到的资料了。
- 本书在28章中讨论了28个不同的主题。这意味着你可以在一本书中找到广泛的Java主题。
- 本书是以参考形式编写的，每章都可以独立阅读。我们尝试在多章中使用类似的例子，但是即使你最先阅读其中的任何一章，你也能够理解其中的说明。

本书适合的读者

本书是为中级Java程序员所写的，所以假设你已经了解Java的基本知识，而且正在寻找比语法更高级的知识。我们不解释与书中主题无关的代码特性。

如果一个程序员具有5年以上的经验（其中至少包含2年使用Java的经历），那么他应该能够毫无问题地理解本书中的解释和例子。经验较少的程序员也能从本书受益，但是当他们无法理解例子时应该查阅初级资料。

阅读方式

本书涉及的范围很广泛，扫一眼本书的目录你就会发现本书讨论了许多主题。因此，讨论

每个主题的篇幅就比较有限了。所以，每一章的目标是帮助你开始研究每个主题。

本书讨论的主题都是本着“使Java程序员的职业技能提高一个层次”的目的而选择的。如果你了解Java，就能够理解本书中的所有例子。如果你分析这些例子、修改它们并且观察发生的现象，你就会发现你能够在会议和谈话中提供关于这些主题的出色建议。本书中的每一章都先提供几页介绍和背景知识，然后是几个使用此技术的例子以及对输出的说明。

没有什么可以替代对例子进行研究。在参加需要你提出建议的会议之前，你应该拿起这本书，花两小时研究几个例子。这样，当你提出一个技术方案时，你的建议更容易被采纳。如果说一张图抵得上千言万语，一个例子的价值则更大。

大多数程序员在换工作时都会带着他们最喜欢的Java入门书和一本高级Java编程书籍。我们希望本书能成为你的第三本案头书。

本书的组织结构

本书分为六个部分：

- 第一部分——分布式对象编程
- 第二部分——企业级Java
- 第三部分——Java集成技术
- 第四部分——Java媒体技术
- 第五部分——设计Java类的最佳方式
- 附录

每个部分的内容都在使你成为更好、更受尊敬的程序员方面扮演重要角色。

第一部分——分布式对象编程

我们的许多工程要求跨越计算机之间的边界。此部分各章解释了应该怎么做。

第1章“计算机互连编程原理”讨论将两台计算机连接在一起的问题。它概述了可以用于实现此目标的技术以及一些历史资料，以便帮助你理解这些技术的起源。它还解释了Transmission Control Protocol/Internet Protocol（传输控制协议/网络互连协议，TCP/IP）是如何工作的。这很重要，因为所有其他技术都是在TCP/IP之上建立的。

第2章“使用RMI创建应用程序”讲解如何使用RMI。它的重要性在于许多分布式技术是在RMI之上建立的。

第3章“使用JAXP创建应用程序”解释了Java如何处理XML。本书描述的许多技术将XML文件作为数据传递。

第4章“使用JSP和servlet进行应用程序开发”讲解如何使用JavaServer Pages和servlet创建系统。对于许多应用程序，servlet提供了简单而且高效的方案。

第二部分——企业级Java

如果你需要可伸缩性，就需要使用Enterprise Java。此部分各章讲解J2EE套件中的先进技术并且帮助你决定其中哪种技术最适合你的需要。

第5章“Enterprise JavaBean (EJB)”讨论当今最热门的技术，EJB。它提供了引导你入门的例子，并且讨论了会话bean和实体bean。

第6章“消息驱动的bean (MDB)”讨论一种崭新的EJB，消息驱动的bean。这种bean的可伸缩性很像EJB，但是通信方式很像消息队列。

第7章“Java消息服务 (JMS)”讨论JMS这种成熟的技术，它为某类企业系统提供了合适的方案。许多JMS应用程序已经被用于在不同类型的编程环境之间进行通信。

第8章“选择正确技术的方法”对比了各种可用技术并且指导你做出明智的选择，包含评估一种技术是否符合你的需要的标准。

第三部分——Java集成技术

Java是一种强大的“粘合”技术。此部分各章讲解程序员可以使用的各种“胶水”。

第9章“Web服务”解释如何使用这种新技术创建通过XML文件进行通信的应用程序。

第10章“JDBC和数据库”讲解如何在Java程序中与数据库进行交互。它还讲解了如何进行批量更新。

第11章“Java命名和目录接口 (JNDI)”解释了Java程序设计中最神秘的一个方面。它揭示了幕后的情况，展示了命名服务的实际工作方式。

第12章“使用Java处理语音”用例子解释了如何在程序中进行语音合成和识别。随着便携设备的普及，这种能力变得越来越重要了。

第13章“JavaMail”讲解如何使用JavaMail协议创建不必面向电子邮件的异步系统。

第14章“串行端口通信”讲解如何使用Java通过计算机上的串行端口和Java Communications Standard Extension与设备进行通信。

第四部分——Java媒体技术

Java是一种很适合处理媒体的平台。此部分各章讲解Java Media包中的功能。

第15章“使用Java 2D进行绘制”向你介绍这个非常重要的API并且提供使用它的例子。这些例子还演示了如何实现2D动画。

第16章“Java 3D图形”讲解如何在3D空间中创建物体。同时还讲解如何使物体运动和加上光线。

第17章“Java媒体框架 (JMF) 和Java Advanced Imaging (JAI) API”讲解如何使用Java播放视频。它还讲解如何使用Java进行图像操作。

第18章“使用Java处理声音”讲解如何在Java程序中捕捉和播放高质量的声音。声音可以采用取样或乐器数字接口 (Musical Instrument Digital Interface, MIDI) 文件的形式。

第五部分——设计Java类的最佳方式

资深开发人员需要理解某些概念。此部分各章集中讨论这些在程序设计团体中还没有被广泛理解的概念。

第19章“JavaBean和Java事件模型”讲解如何创建JavaBean。它还详细解释了在Java类之间如何传递事件。

第20章“体系结构模型和框架”解释如何为你的应用程序创建框架。它还讨论了一些商业框架以及使它们获得成功的原因。

第21章“Java设计模式”讲解如何使用设计模式改进你的系统，使它们更容易维护。对许多较高级设计的讨论要求你理解这些模式。

第22章“异常和调试”讲解如何使用一种全面的方式处理异常。它描述了常见的问题。

第23章“Java安全”讲解如何避免系统中的信息被盗和被破坏。

第24章“Java多线程应用程序”讲解如何创建并行运行的应用程序。许多Java多媒体程序设计都广泛地使用了线程。

第25章“Java国际化”讲解如何创建可以支持多种自然语言的应用程序。它还说明了如何提供变化的图形。

第26章“持久化和克隆”讨论了与串行化相关的问题。这很重要，因为大多数企业技术依赖于串行化对象。

第27章“多态性和继承”解释了Java的继承机制是如何工作的。这对于理解标准扩展是很重要的。

第28章“代码的组织和部署”提供了JAR文件、包和Javadoc的语法。

附录

附录A包含本书的多章中使用的代码。附录B解释如何得到WebLogic Server，这是本书中多次使用的基于J2EE的应用服务器。

本书中使用的约定

本书中使用了以下的排版约定：

- 在显示输入和输出行时，常常使用**粗体**字型表示用户的输入。
- 如果代码行太长，在书中无法放在一行内，就使用一个特殊图标（➡）。如果你看到代码行前面有➡图标，就要记住它是上一行的一部分。

本书还使用注意、警告和提示。这些特殊部分与一般文字分离，它们提供了相关问题的附加信息。

注意 注意用于指出你可能需要这些附加信息来理解文中讨论的概念。因为这些信息的重要性，它们应该特殊对待。

警告 警告用于提醒你注意与文中解释的主题相关的潜在问题。

提示 提示用于给出一般很难得的额外信息。这种信息常常是作者从工作经历中总结出来的。

网站

可以在Sams出版社的网站（<http://www.sampublishing.com/>）上找到本书各章中描述的代码文件。在Search框中输入不加连字符的本书的ISBN号（067232394x）并且点击Search。当本书的标题出现时点击它，你就会进入一个可以下载所有代码的页面。

联系方式

作为作者，我们欢迎我们的读者提出意见和建议。如果你想称赞、批评或纠正本书的内容，或者对我们以后的书提出建议，那么请与我们联系。可以通过电子邮件地址 stevepotts@mindspring.com 与Stephen Potts联系。可以通过电子邮件地址 pestrikov@yahoo.com 与Alex Pestrikov联系。

关于作者

Stephen Potts是美国乔治亚州亚特兰大的一位独立顾问、作家和Java讲师。Stephen于1982年从Georgia Tech获得了计算机科学学位。在20年的职业生涯中，他经受了許多考验，并且成为其研究领域经验丰富的专家。他以前的著作有《Special Edition Using Visual C++ 4》和《Java 1.2 How-To》。可以通过电子邮件地址stevepotts@mindspring.com与他联系。

Alex Pestrikov是一位专攻J2EE技术的独立顾问。他毕业于康涅狄格州的University of Bridgeport。Alex参与了美国和加拿大的许多J2EE工程，最近为Ontario政府提供咨询。他与未婚妻Stephanie在Toronto生活，可以通过电子邮件地址pestrikov@yahoo.com与他联系。

Mike Kopack是Georgia Institute of Technology的毕业生，他在那里获得了计算机科学学士学位。Mike是Sun Certified Java Programmer，他对Java的研究从JDK 1.0.2就开始了。他专攻服务器端Web技术。他早先的工作涉及为大型公司建立动态网站内容管理系统。目前他是乔治亚州亚特兰大的ISX公司的软件工程师。

目 录

前言

关于作者

第一部分 分布式对象编程

第1章 计算机互连编程原理	1
1.1 计算机连接简史	1
1.2 TCP/IP和Java	3
1.2.1 TCP/IP工作原理	3
1.2.2 应用层	5
1.2.3 传输层	6
1.2.4 网络层	6
1.2.5 网络访问层	6
1.3 理解现代互连技术	10
1.3.1 Java数据库连接性 (JDBC)	11
1.3.2 TCP/IP	11
1.3.3 servlet	11
1.3.4 远程方法调用 (RMI)	12
1.3.5 Java命名和目录接口 (JNDI)	12
1.3.6 JavaMail	12
1.3.7 Java消息服务 (JMS)	13
1.3.8 Enterprise JavaBean (EJB)	13
1.3.9 消息驱动的bean (MDB)	13
1.3.10 XML	14
1.3.11 Web服务	14
1.4 旅行票据应用程序的设计	14
1.5 小结	15
第2章 使用RMI创建应用程序	16
2.1 理解RMI	16
2.1.1 RMI特性	17
2.1.2 RMI与CORBA的比较	17
2.1.3 RMI的内部机理	18

2.1.4 支持程序	18
2.1.5 创建RMI应用程序	19
2.1.6 运行示例程序	25
2.2 将应用程序转换为RMI	25
2.3 创建CruiseList应用程序的RMI版本	37
2.4 小结	54
第3章 使用JAXP创建应用程序	55
3.1 理解XML	55
3.2 XML相关工具	56
3.3 创建XML文档	57
3.4 创建文档类型定义 (DTD)	59
3.5 使用Simple API for XML (SAX) 进行分析	61
3.6 使用文档对象模型 (DOM) 进行分析	66
3.7 CruiseList应用程序的XML版本	72
3.8 小结	92
第4章 使用JSP和servlet进行应用 程序开发	93
4.1 HTTP基础	93
4.2 第一个JSP页面	94
4.3 JSP标记类型	98
4.3.1 指令	98
4.3.2 声明	99
4.3.3 表达式	100
4.3.4 JSP小脚本	100
4.4 使用JSP处理表单	101
4.5 第一个servlet	105
4.6 何时使用JSP, 何时使用servlet	108
4.7 综合使用JSP和servlet	109
4.8 小结	128

第二部分 企业级Java

第5章 Enterprise JavaBean (EJB)129

5.1 理解EJB方式130

5.1.1 EJB的长处131

5.1.2 EJB的弱点131

5.1.3 编写简单的EJB132

5.2 EJB内部原理139

5.2.1 EJB体系结构139

5.2.2 实例缓存140

5.2.3 部署说明文件概述141

5.2.4 EJB的类型141

5.3 使用EJB开发旅行订票应用程序143

5.4 开发无状态会话bean144

5.4.1 使用无状态会话bean进行 实例缓存144

5.4.2 无状态会话bean的状态图144

5.4.3 无状态bean中的回调方法145

5.4.4 为旅行订票应用程序编写 无状态bean145

5.4.5 无状态会话bean中的create <METHOD>方法151

5.5 开发有状态会话bean151

5.5.1 有状态会话bean的状态图152

5.5.2 SessionSynchronization接口152

5.5.3 有状态bean中的回调方法152

5.6 开发实体bean153

5.6.1 实体bean中的持久化模型153

5.6.2 实体bean的状态图154

5.6.3 实体bean中的回调方法154

5.6.4 为旅行订票应用程序编写 实体bean154

5.6.5 实体bean中的create<METHOD> 方法162

5.6.6 EJB查询语言162

5.6.7 find<METHOD>方法163

5.6.8 ejbSelect<METHOD>方法163

5.6.9 ejbHome<METHOD>方法163

5.6.10 CMP字段163

5.6.11 CMR字段164

5.6.12 主键164

5.7 建立并且运行旅行订票应用程序165

5.8 EJB容器服务166

5.8.1 EJB中的声明式安全166

5.8.2 EJB中的声明式事务167

5.8.3 EJB环境169

5.9 小结169

第6章 消息驱动的bean (MDB)171

6.1 理解消息驱动的bean (MDB)172

6.2 开发和部署MDB173

6.2.1 编写MDB Java文件173

6.2.2 手工创建部署说明文件173

6.2.3 部署消息驱动的bean175

6.2.4 编写简单的MDB176

6.3 MDB的工作原理183

6.4 点对点模型184

6.5 创建基于GUI的应用程序192

6.6 在MDB中处理错误211

6.7 小结212

第7章 Java消息服务 (JMS)213

7.1 理解JMS214

7.2 使用JMS建立旅行票据系统214

7.2.1 用户界面编程215

7.2.2 使用JMS建立票据代理应用程序227

7.2.3 集成测试236

7.3 发布/预订模型236

7.4 小结246

第8章 选择正确技术的方法247

8.1 选择技术的挑战247

8.2 衡量软件特征248

8.2.1 成本248

8.2.2 风险249

8.2.3 可维护性249

8.2.4 可靠性249

8.2.5 可伸缩性	249	9.6 与.NET进行集成	286
8.2.6 互操作性	250	9.7 小结	288
8.2.7 简单性	250	第10章 JDBC和数据库	290
8.2.8 可扩展性	250	10.1 为什么需要JDBC	290
8.2.9 可检验性	250	10.2 JDBC的工作原理	291
8.2.10 人员的情况	250	10.3 批处理	293
8.2.11 学习的难度	251	10.4 JDBC与WebLogic Server	298
8.3 评估分布式技术	251	10.5 JDBC事务管理	304
8.3.1 Java和Java数据库连接性 (JDBC)	251	10.6 JDBC 3.0中的改进	316
8.3.2 TCP/IP编程(套接字)	252	10.7 小结	317
8.3.3 servlet	253	第11章 Java命名和目录接口(JNDI)	318
8.3.4 远程方法调用(RMI)	254	11.1 为什么需要命名服务API	319
8.3.5 Java命名和目录接口(JNDI)	254	11.2 理解JNDI	319
8.3.6 JavaMail	256	11.3 命名服务的工作原理	320
8.3.7 Java消息服务(JMS)	256	11.4 寻找JNDI提供者	322
8.3.8 消息驱动bean(MDB)	257	11.4.1 默认寻找	322
8.3.9 Enterprise JavaBean(EJB)	258	11.4.2 在命令行上传递属性	324
8.3.10 Web服务	259	11.4.3 创建JNDI属性文件	324
8.4 对比Java技术与CORBA	259	11.4.4 传递属性对象	324
8.5 小结	260	11.4.5 创建Hashtable	326
第三部分 Java集成技术		11.5 JNDI与WebLogic Server	328
第9章 Web服务	261	11.5.1 绑定、取消绑定和重新绑定	330
9.1 理解Web服务方式	262	11.5.2 复杂对象	332
9.1.1 什么是Web服务	262	11.6 使用JNDI时的注意事项	340
9.1.2 Web服务的长处	263	11.7 小结	340
9.1.3 Web服务的弱点	264	第12章 使用Java处理语音	341
9.1.4 创建一个简单的Web服务	265	12.1 理解Java Speech	341
9.2 理解SOAP	271	12.2 创建和分配语音引擎	343
9.3 创建一个付款Web服务	274	12.2.1 引擎状态	345
9.3.1 SOAP错误消息	278	12.2.2 分配线程	348
9.3.2 SOAP数据类型	280	12.3 语音合成	350
9.4 理解WSDL	281	12.4 语音识别	358
9.5 理解UDDI	284	12.5 小结	362
9.5.1 UDDI数据结构	284	第13章 JavaMail	363
9.5.2 UDDI API	285	13.1 理解JavaMail	363
		13.1.1 Session类	364
		13.1.2 Message类	364

13.2	JavaMail的工作原理	364	17.3	使用Advanced Imaging API进行 程序设计	478
13.3	创建简单的SMTP应用程序	366	17.4	小结	482
13.4	创建基于JavaMail的CruiseList 应用程序	374	第18章	使用Java处理声音	484
13.5	获取特定的电子邮件消息	397	18.1	取样声音的原理	484
13.6	编写事件驱动的电子邮件应用程序	402	18.2	MIDI声音的原理	485
13.7	小结	407	18.3	Java Sound API	486
第14章	串行端口通信	408	18.4	使用Java播放声音	488
14.1	理解RS-232标准	408	18.5	使用Java捕捉声音	498
14.1.1	RS-232硬件	409	18.6	小结	510
14.1.2	通信软件	410			
14.2	Java Communications API	411	第五部分	设计Java类的最佳方式	
14.3	通过串行端口发送数据	419	第19章	JavaBean和Java事件模型	511
14.4	小结	425	19.1	理解JavaBean	512
			19.2	使用Java事件模型进行程序设计	512
			19.3	开发简单的JavaBean	517
			19.4	方法、事件和属性	527
			19.4.1	方法	527
			19.4.2	事件	529
			19.4.3	绑定属性和受限属性	529
			19.5	小结	534
			第20章	体系结构模型和框架	535
			20.1	为什么需要框架	535
			20.2	什么是框架	536
			20.3	MVC模式概述	540
			20.4	Jakarta Struts框架中的MVC实现	542
			20.4.1	在Struts中创建控制器	544
			20.4.2	在Struts中创建视图	547
			20.4.3	在Struts中创建模型	552
			20.4.4	运行登录应用程序	552
			20.5	小结	553
			第21章	Java设计模式	554
			21.1	为什么需要模式	554
			21.2	Java模式	555
			21.2.1	单实例	555
			21.2.2	枚举	557
			21.2.3	工厂	560

第四部分 Java媒体技术

第15章 使用Java 2D进行绘制 427

15.1 理解Java 2D 427

15.2 在屏幕上绘制文本 428

15.3 绘制图形 433

15.4 操作图像 437

15.5 混合使用文本、图形和图像 443

15.6 小结 445

第16章 Java 3D图形 447

16.1 理解三维(3D)编程 447

16.2 Java 3D API中的主要对象 449

16.3 在Java 3D应用程序中添加运动 453

16.4 在Java 3D程序中添加光线 454

16.5 在Java 3D中添加纹理映射 460

16.6 小结 465

第17章 Java媒体框架(JMF)和Java Advanced Imaging(JAI) API 466

17.1 理解基于时间的媒体 466

17.2 使用JMF对基于时间的媒体
进行解码 468

17.2.1 添加控制面板 473

17.2.2 JMF中的其他功能 478

21.2.4 观察者	563	24.1.1 线程的生存周期	638
21.2.5 代理	567	24.1.2 创建线程	639
21.2.6 外观	571	24.2 控制线程	640
21.3 EJB设计模式	573	24.2.1 启动线程	640
21.3.1 业务委托	573	24.2.2 使线程睡眠	641
21.3.2 命令	578	24.2.3 中断线程	642
21.4 小结	582	24.2.4 停止线程	644
第22章 异常和调试	583	24.3 创建线程安全的应用程序	644
22.1 异常处理的重要性	583	24.3.1 多线程应用程序中最常见的 问题	644
22.2 理解Java异常策略	585	24.3.2 临界区	646
22.3 在Java程序中捕获异常	585	24.3.3 让线程等待	648
22.4 常见的异常处理方式	593	24.3.4 结合线程	651
22.5 处理异常的正确方式	595	24.4 线程管理	651
22.6 转换异常	598	24.4.1 线程组	651
22.7 传递异常而不损失信息	600	24.4.2 线程的优先级	654
22.8 JDK 1.4中的异常传递	610	24.5 小结	654
22.9 小结	618	第25章 Java国际化	656
第23章 Java安全	619	25.1 国际化的基本原理	656
23.1 Java安全概述	619	25.2 为文本使用资源束	657
23.1.1 应用程序安全综述	619	25.3 使用ResourceBundle类存储对象	666
23.1.2 安全威胁	620	25.4 国际化日期和时间	670
23.1.3 Java中的安全机制	620	25.5 国际化数值和货币值	673
23.1.4 Java Authentication and Authorization Service综述	621	25.6 小结	678
23.1.5 Java Cryptography Extension 综述	621	第26章 持久化和克隆	679
23.2 Java沙箱模型	622	26.1 对象串行化	679
23.2.1 Java沙箱体系结构	622	26.1.1 自动串行化	680
23.2.2 使用Java权限	623	26.1.2 transient关键字	688
23.2.3 为Java沙箱编写自己的权限	626	26.1.3 带版本的串行化	689
23.3 Java安全套接字	629	26.1.4 定制的串行化	691
23.3.1 SSL综述	629	26.1.5 串行化内部原理	693
23.3.2 加密综述	630	26.2 对象外表化	694
23.3.3 使用JSSE的客户/服务器	630	26.3 克隆	697
23.4 小结	636	26.3.1 创建深度副本	703
第24章 Java多线程应用程序	637	26.3.2 使用串行化进行克隆	705
24.1 线程概述	637	26.3.3 AutoCloner与性能	708
		26.4 安全与串行化	708

26.5 小结	714	27.11 小结	738
第27章 多态性和继承	715	第28章 代码的组织和部署	739
27.1 多态性和继承的重要性	715	28.1 理解包	739
27.2 多态性的内部原理	716	28.2 JAR文件	741
27.3 多态性编程	717	28.2.1 内容清单文件	744
27.4 继承、构成或强力	722	28.2.2 运行JAR文件中的代码	744
27.5 继承和多态性的缺点	725	28.2.3 扩展	745
27.6 具体或抽象基类	726	28.3 Java API Documentation Generator	746
27.7 类型安全的向下类型转换与继承	727	28.4 小结	756
27.7.1 instanceof解决方案	729		
27.7.2 Object.getClass() 解决方案	729		
27.7.3 Class.forName() 解决方案	730		
27.8 用运行时类型识别进行多态性编程	732		
27.9 用反射进行多态性编程	735		
27.10 继承与句柄	737		

第六部分 附 录

附录A 本书使用的实用程序源代码 清单	757
附录B 下载和安装WebLogic Server	774

第一部分 分布式对象编程

本部分主要内容：

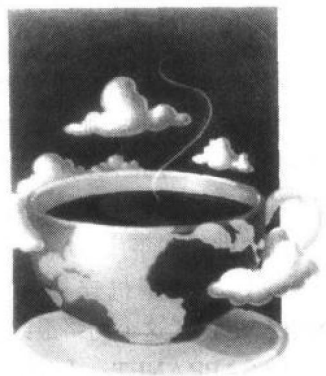
- 第1章 计算机互连编程原理
- 第2章 使用RMI创建应用程序
- 第3章 使用JAXP创建应用程序
- 第4章 使用JSP和servlet进行应用程序开发

第1章

计算机互连编程原理

本章主要内容：

- 计算机连接简史
- TCP/IP和Java
- 理解现代互连技术
- 旅行票据应用程序的设计



自从第二台计算机出现以来，人们就希望将计算机连接起来。仿佛我们需要的数据总是在另一台运行着另一种操作系统的计算机上。在本章中，你将学习能够使计算机相互连接的技术。

首先，我们将简短地介绍计算机连接的历史。接下来我们将看到几个简单的TCP/IP例子。TCP/IP技术是其他所有连接技术的基础，所以理解这些例子有助于你阅读那些讨论在TCP/IP之上建立的技术（比如RMI、EJB、JNDI和JMS）的章节。

然后，我们将在较高层次上了解当前用于连接计算机和交换信息的技术。第8章将对各种技术，以便帮助程序员选择最适合特定需求的技术。本章只是概述这些技术，在后续几章中将讨论细节。

最后，我们将看到一个简单的旅行票据系统的设计。后面几章将使用这个系统作为例子。后面几章将给出这个应用程序的细节，这里只概述这个系统。

1.1 计算机连接简史

在计算机时代的早期，计算机非常大，需要很大的房间才能容纳，而且它们只能同时运行一个应用程序；任何时候只能有一个用户运行程序。用于解决问题的所有数据都必须在同一台

机器上。随着计算机能力的增强,可以通过非常快速地切换任务,在一台计算机上执行多个任务。这使多个应用程序可以看起来像是同时运行的。

不久之后,工程师们设计了一种办法,使多个用户可以同时使用同一台计算机运行他们的任务,而且大型机诞生了。大型机是系统的实际大脑,每个用户使用一个终端(通常是一个哑CRT屏幕和键盘或电传机)通过一个串行调制解调器连接大型机。大型机处理和所有内容。如果没有大型机,客户终端就什么事也做不了。

大型机工作得很好,因为所有数据都在一个地方。如果需要处理报表,所需的数据与报表编写程序总是在同一台计算机上。但是大型机也有缺陷。购买和操作这些大型计算机的费用很高。它们需要特殊的冷却系统,还需要一组技术人员监视它们和为它们编程。另外,它们不能很好地支持远程用户。如果你在亚特兰大而你的大型计算机在圣路易斯,那么就需要租用和安装非常昂贵的线路才能实现连接。

另一个问题是互操作性。在不同厂商生产的两台计算机之间进行通信是很困难的。在大多数情况下,大型机厂商不认为使他们的机器能够轻松地连接到其竞争对手的系统会有什么好处。无论大型计算机存在多少缺点,它们毕竟为许多复杂系统提供了平台,一些大型机至今仍在工作。

另一个问题是所有相关数据必须在同一台大型机上(这个问题仍然存在)。如果你需要另一台大型机上的数据,那么通常将数据保存到某些介质上(比如磁带、穿孔卡片或纸带),然后将这些介质搬到你正在使用的机器上。你基本上是使用数据的完整副本,而不是使用原始数据,所以对原始数据的更新并不会对副本起作用,反之亦然。这会带来许多麻烦。

在20世纪60年代,Advanced Research Projects Agency (APRA)启动了一个工程,它的目的是找到将大型机连接起来的方式,以便共享数据。这导致建立了第一个真正的网络(APRANET),这个网络最终发展为当今的Internet。

有了APRANET,大型机终于能够共享数据,而无须用户在机器之间搬运物理数据带或卡。电子邮件等应用程序产生了,第一个数据库也出现了,而且大型机上的用户可以访问另一台大型机上的文件了。

随着这些革新出现了一系列联网技术,因此客户/服务器方式逐渐开始流行了。小公司需要比购置和运行大型机更便宜的解决方案。小公司和公司中需要更多控制计算资源的部门开始青睐Digital Equipment、Sun Microsystems和Hewlett-Packard等公司生产的较小型的计算机。

这些厂商在技术上非常先进,而且他们为争夺市场份额进行了激烈的竞争。他们提供了远远优于大型机的连接能力。Ingres和Oracle等厂商生产的数据库管理系统使用户能够从不同的计算机访问他们的数据。数据库被放在服务器上,而包含业务逻辑的程序被放在客户机上。这就是“客户/服务器(client/server)”这个术语的来源。

这种方式解决了大型机的一些问题。服务器的价格更便宜,而且它们可以连接到比它们还便宜的客户机,比如X Window终端或个人计算机。客户/服务器系统的主要缺点是难以维护客户机上的代码。在这种模型中,如果有100台客户机访问同一台服务器,那么对于客户软件的每次升级和修补都需要执行100次安装。无论如何,许多成功的系统是使用这种模型编写和部署的。

为了解决客户/服务器方式导致的问题,工程师们开发了三层方式。在三层方式中,在客户上处理数据的表示,在服务器上存储和处理数据。这种方式还包含另一个服务器(或层),它只