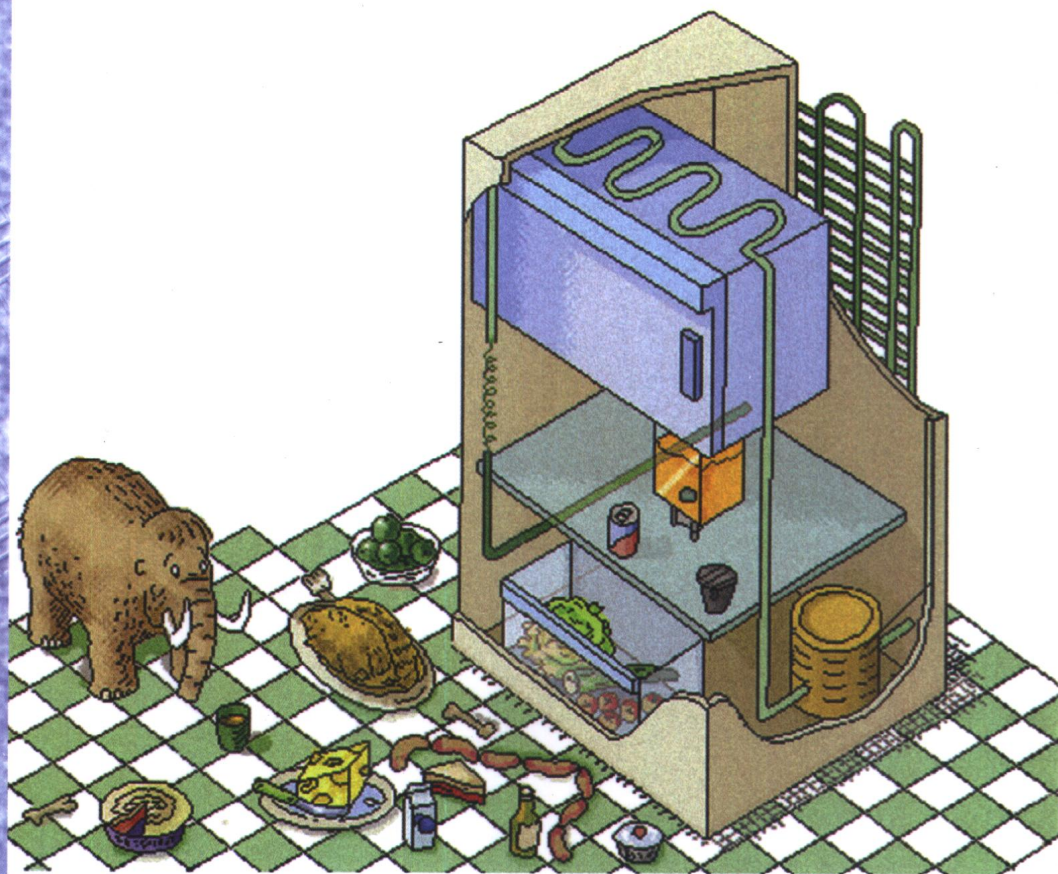


J2EE Unleashed

Sun 公司核心技术丛书

J2EE

技术内幕



(美) Joseph J. Bambara 等著
Paul R. Allen

刘 堃 等译



机械工业出版社
China Machine Press

SAMS

876

TP3125A
B11

Sun公司核心技术丛书

J2EE技术内幕

Joseph J. Bambara Paul R. Allen

(美) Mark Ashnault Ziyad Dean 著

Thomas Garben Sherry Smith

刘 堃 等译



A0979695



机械工业出版社
China Machine Press

本书从Java技术的背景和企业系统开发过程开始谈起,然后着眼于J2EE应用开发中需要做的工作,并谈到了一些关键技术。包括Java命名和目录接口、轻型目录访问协议、Java数据库连接、Java服务器端小程序(servlet)、Java服务器页面(JSP)、模型-视图-控制器(MVC)和企业JavaBeans(EJB)。然后又论述了J2EE应用的专用组件,主要集中在JavaMail、Java消息服务(JMS)和可扩展标记语言(XML)。最后还讨论了启用和部署程序所涉及的许多问题。

全书包括19章及三个附录,内容非常全面实用,是企业构建基于Web的应用和分布式应用时非常宝贵的参考资源。

Joseph J. Bambara, Paul R. Allen et al: J2EE Unleashed.

Authorized translation from the English language edition published by Sams, an imprint of Macmillan Computer Publishing U.S.A.

Copyright © 2002 by Sams Publishing.

All rights reserved.

Chinese simplified language edition published by China Machine Press, Copyright © 2002 by China Machine Press.

本书中文简体字版由美国麦克米兰公司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:01-2002-0812

图书在版编目(CIP)数据

J2EE技术内幕 / (美)巴姆巴拉(Bambara, J.J.)等著;刘垚等译. -北京:机械工业出版社, 2002.6

(Sun公司核心技术丛书)

书名原文: J2EE Unleashed

ISBN 7-111-10200-2

I. J... II. ①巴... ②刘... III. Java语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字(2002)第029326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:李敏杰 刘立卿

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2002年6月第1版第1次印刷

787mm×1092mm 1/16·37.5印张

印数:0 001-4 000册

定价:65.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

译者序

Java 2平台的企业版(J2EE, Java 2 Enterprise Edition)定义了开发多层企业应用程序的标准。就像SQL一样, J2EE通过提供标准化、模块化的组件, 并为这些组件提供完整的服务, 以及自动处理应用程序行为的许多方面, 简化了企业应用程序的开发。

Java 2平台的企业版具有标准Java 2的许多特色优势, 例如“Write Once, Run Anywhere”(编写一次, 随处运行)的可移植性、数据库访问的JDBC API、同现有企业资源进行交互的CORBA技术。在这个基础上, J2EE添加了对企业JavaBeans组件、Java Servlet API、JavaServer页面和XML技术的支持。J2EE标准包括完整的规范和一致性测试, 确保应用程序可以运行在大多数支持J2EE的现有企业系统中。这种可移植性正是SQL成功的关键因素。

本书从Java技术的背景和企业系统开发过程开始谈起。然后着眼于J2EE应用开发中需要做的工作, 并谈到了一些关键技术。最后几章讨论了启用和部署程序所涉及的许多问题。附录中包含了支持本书内容的参考资料(包括配置本书使用的商业和范例软件的说明), 并提供了可以用来扩展本书内容的其他参考信息。在本书中, 作者尝试在多层面上讨论J2EE, 为使用J2EE构建基于Web的和分布式企业系统提供操作性很强的指导。本书的作者都是纽约金融部门计算机解决方案供应商的顾问, 平均有12年的开发经验。从根本上来说, 本书是非常实用的, 是企业构建基于Web的应用和分布式应用时非常宝贵的参考资源!

现在, 我们很荣幸能够有机会承担本书的翻译工作。在翻译过程中, 我们经常为一句话、一个术语进行反复的讨论, 到处查找资料, 力图使本书的翻译能正确、贴切地反映原文的意思, 同时注意使句子、段落符合中国人的语言习惯。我们真挚地希望读者能够从本书中有所收获, 这是作者的初衷, 也是我们的愿望!

本书由刘堃等组织翻译, 万方工作室的全体同仁都参加了本书的翻译、校正和输入等工作。具体参加本书翻译、录排、校对工作的其他人员为: 刘堃、潘德宏、田蕴哲、丁天、牛志奇、王小将、刘砚、黄建春、刘胜利、蒋雯丽、葛丽、罗贤锋、罗天浩、王若宾、潘文凯、夏宣哲、李树玲、孙志楠、聂宛析、龚露娜、马志强、李秀芬、田野、牛献忠、金百湘、薛杰、叶欣哲、邓南燕、邢倩、王煜、李满军、刘思敏、钱凯、赵智策、江南、李昌、王霄、李林等。本书的出版是集体劳动的结晶, 在此特别感谢万方工作室的全体工作人员。

由于时间仓促, 且译者经验和水平有限, 译文难免有不妥之处, 恳请读者批评指正!

万方工作室

2002年1月

前 言

使用J2EE开发应用程序同开发其他类型的应用程序非常相似，但却有自己独特的优势。相似是因为开发过程中的迭代步骤是相同的；真正的优势是指J2EE提供了过程标准，就像数据库标准SQL一样。

J2EE标志着技术成为标准的时刻。同数据库开发中的SQL一样，软件开发供应商大多支持J2EE作为开发企业应用程序的标准。

本书讨论范围

本书的作者都是纽约金融部门计算机解决方案供应商的顾问，平均有12年的开发经验。随着技术发展，我们需要不断汲取新技术，并应用之。

在本书中，我们尝试在多层面上讨论J2EE，为使用J2EE构建基于Web和分布式企业系统提供操作性很强的指导。

本书的组织结构

本书从Java技术的背景和企业系统开发过程开始谈起。然后着眼于J2EE应用开发中需要做的工作，并谈到了一些关键技术，包括Java命名和目录接口（JNDI, Java Naming and Directory Interface）、轻型目录访问协议（LDAP, Lightweight Directory Access Protocol）、Java数据库连接（JDBC, Java Database Connectivity）、Java服务器端小程序（Java Servlet）、Java服务器页面（JSP, JavaServer Pages）、模型-视图-控制器（MVC, Model-View-Controller）和企业JavaBeans（EJB）。然后又论述了J2EE应用的专用组件，主要集中在JavaMail、Java消息服务（JMS, Java Message Service）和可扩展标记语言（XML, eXtensible Markup Language）。最后几章讨论了启用和部署程序所涉及的许多问题。附录中包含了支持本书内容的参考资料（包括配置本书使用的商业和范例软件的说明），并提供了其他参考信息以扩展所学内容。每章内容概述如下：

第1章为“Java、分布式计算和J2EE”。从Java的背景谈起，谈到了当前Web应用开发的需求，也提到了J2EE规范和认证考试，这些可以验证供应商平台的兼容性。本章概要讨论了J2EE应用程序的打包方式，介绍了本书后面章节要研究的范例应用程序。

第2章为“J2EE应用设计和开发”。从J2EE结构总体开始谈起，进行了开发方法和过程的分析。本章讨论了开发环境，其中包括当前可用的各种集成开发环境（IDE, Integrated Development Environment）。本章还讨论了J2EE应用程序安全方面的问题，以及在设计、创建、组装和部署J2EE应用程序时需要的各种角色。

第3章为“设计J2EE应用”。首先讨论了建模工具的历史，其中建模工具包括数据库、过程和对象建模工具。然后解释了建模的重要性，以及建模和Java如何完美结合。本章着眼于如何从建模中生成EJB和其他类型的类，还解释了范例应用程序中使用的模型。

第4章为“构建J2EE应用程序的任务列表”。任务列表可以用在J2EE开发项目的起始阶段。本章包含了建立对数据库的访问，以及开发数据定义语言 (DDL, Data Definition Language) 和数据操作语言 (DML, Data Manipulation Language) 的各个步骤。还包含了构建应用程序的业务逻辑和数据库访问的各个步骤，包括工具类、会话bean和实体bean。它还包含导航和应用程序流的映射，例如Struts、XML文件、servlet和JSP页面的使用。

第5章为“建立资源访问：JNDI和LDAP”。描述了普通的命名服务，重点讨论了Java命名和目录接口。本章讨论了JNDI的结构和接口，解释并举例演示了分布式应用程序的组件如何使用JNDI相互定位。另外，本章还介绍了目录服务，详细讨论了轻型目录访问协议。

第6章为“建立数据访问：JDBC”。解释了Java数据库连接技术，这是Java平台不可分割的一部分。本章根据Core程序包讨论了JDBC，包括驱动程序类型、数据库连接、处理查询结果、把SQL类型映射到Java，以及获取数据库的相关信息。还讨论了JDBC 2.0可选程序包API，包括使用JNDI的数据库命名机制、连接池、分布式事务、数据集 (RowSet) 和高级数据类型。还介绍了在SilverStream应用服务器中使用JDBC的代码范例，包括SilverStream连接池，以及在SilverStream应用服务器中从会话bean访问JDBC。

第7章为“建立控制流：servlet”。论述了Java Servlet。对构建交互式Web应用来说，Java Servlet是当前最流行的选择。本章介绍了Java Servlet，讨论了对于企业Web应用开发和部署来说，servlet在J2EE框架中所处的角色。本章介绍了Servlet API，集中讨论如何构建Java Servlet来产生HTTP请求的响应。

第8章为“Java服务器页面：介绍”。介绍了Java服务器页面，从解释它们的含义和所处结构开始，覆盖了可以使用JSP的各种模型，包括模型1和模型2。并结合Java服务器页面关联的视图部分讨论了模型-视图-控制器结构。本章还把JSP同Java Servlet做了一番比较。

第9章为“Java服务器页面：使用标签库进行实际开发”。讨论了JSP页面的语法，包括对象作用域、隐含创建的对象、注释、声明、表达式、脚本和指令。本章包含创建JSP的各个步骤，包括如何使用JavaBean组件。还讨论了异常处理和HTML表单处理，最后讨论了标签库，其中标签库封装了可以在许多页面上重复使用的动作。

第10章为“为应用构建用户界面”。描述了本书使用的范例程序的用户界面部分。本章还提供了与模型-视图-控制器相关的其他信息，以及在Struts标签库中的实现方法。

第11章为“企业JavaBeans：介绍”。从企业JavaBeans和分布式程序设计开始，演示了EJB如何紧密结合到分布式计算结构中，以及如何把对象组织成合适的程序包层次结构。本章介绍了EJB容器、EJB会话bean、EJB 实体bean和EJB消息驱动bean。讨论了EJB的各个属性，包括持久性、状态和事务能力。本章还提及在EJB的创建和部署中涉及的各种角色（开发者、组装者和部署者）。

第12章为“企业JavaBeans：构建会话bean”。集中讨论了构建EJB会话bean中需要的步骤和问题，其中包括会话bean的有状态和无状态实现之间的比较。本章引导读者学习bean的创建和相关的Home和Remote接口。还提供了调用会话bean的范例代码。

第13章为“企业JavaBeans：构建实体bean”。讨论了构建实体bean过程中涉及的步骤和问题。其中包含定义实体bean、创建主关键字类、创建实体bean类、实现Home和Remote接口，以

及实现实体bean的方法。本章还比较了bean管理持久性（BMP, bean-managed persistence）与容器管理持久性（CMP, container-managed persistence）之间的区别，讨论了实体bean的打包和部署。

第14章为“使用电子邮件传递信息：JavaMail”。在讨论JavaMail API的细节和主要功能之前，概述了普通的信息传递机制，论述了JavaMail的结构，以及发送和接收邮件需要用到的主要类。在此过程中，本章还讨论了服务供应商、Java启动框架（JAF, Java Activation Framework），以及Message和MimeMessage对象。

第15章为“应用间的消息通信：Java消息服务”。解释了Java消息服务（JMS, Java Message Service），这是为Java开发人员提供的API，可以用来创建、发送、接收和读取企业信息传递系统的消息。本章讨论了消息机制的基础，以及用来生成和销毁消息需要的JMS组件，还提供了许多点对点(point-to-point)和发布/订阅(publisher/subscribe)JMS应用程序的真实范例。

第16章为“使用XML进行数据交换”。讨论了可扩展标记语言的基本原理，包括XML文档的结构、解析后的数据、处理指令、文档类型定义（DTD, Document Type Definition）、模式，以及使用样式表来显示XML数据。还讨论了文档对象模型（DOM, Document Object Model）、XML简化API（SAX, Simple API for XML），以及这两种格式的解析器。

第17章为“验证应用”。讨论了测试的基本原理，包括问题的判断方法和解决方案，查找问题时应该考虑的方面，以及一些可以用来测试软件的工具。

第18章为“执行应用”。讨论了应用程序的性能和调整问题，从执行调整的准备阶段开始，包括了诊断和修复性能问题指南，还为Java Servlet、JSP和EJB提供了一些提高性能的提示。还讨论了影响Java性能的因素，以及应该避免使用哪些Java语法。最后讨论了数据库监控和调整工具的范例。

第19章为“部署应用”。讨论了把J2EE应用程序部署到J2EE应用服务器的各个步骤。其中讨论了Java的WORA观点，并讨论了全部存档文件类型，包括JAR、WAR、CAR和EAR。还讨论了XML部署描述符和XML部署计划，这两个方面被用来定义资源引用和XML部署计划，其中XML部署计划被用于解析资源引用。并介绍了部署J2EE应用程序的工具，比较了当前部分J2EE应用服务器。

附录A为“示例应用的文档”。提供了本书使用的范例程序的信息和文档。

附录B为“相关工具”。包含了第三方软件供应商和相关工具的信息。

附录C为“快速参考资料”。可以作为参考指南，其中包括J2EE API参考、J2EE安装、命名规范和其他辅助资料。

告诉我们你的想法

作为本书的读者，你是我们最重要的批评家和评论员。我们重视你的建议，并企盼得知我们的优点、需要改善的地方、读者期望阅读的图书领域，以及其他你希望传达给我们的任何观点。

来信时请确保注明了本书的标题和作者的名字，以及你的名字和电话号码或传真号码。我会仔细阅读你的评价，并同本书的作者和其他编辑分享这些内容。

目 录

译者序

前言

第1章 Java、分布式计算和J2EE1

1.1 Java背景知识4

1.1.1 应用开发的各项能力4

1.1.2 Java是应用开发的粘合剂5

1.2 公司必须为网络经济进行改造6

1.2.1 公司日益需要Internet来参与竞争6

1.2.2 企业应用开发的挑战7

1.3 Web结构的要求9

1.3.1 速度的比较9

1.3.2 服务有效性10

1.3.3 连接到现有数据10

1.3.4 扩展的用户概念: 客户、职员和
合伙人10

1.3.5 灵活的用户交互10

1.3.6 灵活的事务组件模型11

1.4 Web应用的生存周期11

1.4.1 多个开发人员角色11

1.4.2 迭代式开发12

1.4.3 简化的结构和开发13

1.4.4 方便地映射到应用功能13

1.4.5 基于组件的结构14

1.4.6 对客户端组件的支持14

1.4.7 对事务逻辑组件的支持15

1.5 J2EE API和认证16

1.5.1 Java 2 平台企业版规范17

1.5.2 J2EE兼容性测试套件17

1.5.3 J2EE中的Sun BluePrints设计指南18

1.5.4 将来的企业标准18

1.6 XML和J2EE18

1.6.1 为什么使用XML19

1.6.2 电子数据交换和电子商务19

1.6.3 电子数据交换20

1.6.4 企业应用集成20

1.6.5 软件开发和XML20

1.6.6 XML技术和Java平台21

1.6.7 Java 平台的XML技术标准扩展21

1.7 J2EE应用的打包22

1.8 本书研究的应用范例22

1.9 小结23

第2章 J2EE应用设计和开发24

2.1 J2EE层次24

2.1.1 客户层25

2.1.2 表示层25

2.1.3 业务逻辑层25

2.1.4 数据层26

2.2 J2EE应用组件26

2.2.1 客户组件26

2.2.2 Web组件28

2.2.3 业务组件28

2.3 J2EE结构29

2.3.1 容器和服务29

2.3.2 容器类型30

2.4 开发的方法和过程30

2.4.1 建模工具31

2.4.2 开发工具31

2.4.3 J2EE应用的内容32

2.4.4 J2EE应用的开发阶段33

2.5 范例应用程序简介38

2.6 小结38

第3章 设计J2EE应用39

3.1 建模工具的历史39

3.2 概况39

3.3 设计应用	39	4.5 创建后端接口	71
3.3.1 为什么创建模型	40	4.5.1 初始数据的加载	71
3.3.2 实体关系基础	41	4.5.2 创建EJB类和数据库存储过程	72
3.3.3 CASE工具	43	4.5.3 数据库调整和修复的批处理工具	72
3.3.4 逆向工程	44	4.5.4 备份和恢复的批处理工具	72
3.3.5 实体建模	46	4.6 创建界面	72
3.3.6 关系建模	47	4.6.1 选择应用的用户界面风格	73
3.3.7 支持的数据库	49	4.6.2 为界面风格设置TAGLIB类库	73
3.4 为Web应用建模	49	4.7 构建页面	74
3.5 系统建模	51	4.7.1 确定页面类型	74
3.6 统一建模语言	52	4.7.2 为页面或子页面添加控件	75
3.6.1 UML的目标和原则	53	4.7.3 设计菜单交互动作	76
3.6.2 规范和术语	53	4.8 创建数据访问对象	76
3.7 为Web页面建模	56	4.9 验证代码	76
3.7.1 表单	59	4.9.1 决定使用调试器的时机	76
3.7.2 框架	59	4.9.2 选择断点代码	77
3.8 EJB UML映射	59	4.10 提炼代码	77
3.8.1 Java社团应该遵循规范	60	4.11 创建可执行程序	77
3.8.2 将要开发的规范, 以及它如何满足 当前的需求	60	4.12 小结	78
3.9 工具支持	61	第5章 建立资源访问: JNDI和LDAP	79
3.10 小结	61	5.1 命名和目录服务	79
第4章 构建J2EE应用程序的任务列表	62	5.2 什么是JNDI	79
4.1 完成前提任务	62	5.3 寻找应用范例资源	80
4.1.1 后勤前提条件	62	5.3.1 JNDI的好处	80
4.1.2 物理前提条件	64	5.3.2 JNDI和J2EE	80
4.2 设计数据库	65	5.3.3 JNDI服务提供者	81
4.2.1 确定应用实体	65	5.4 JNDI结构	82
4.2.2 提炼每个实体和属性	66	5.4.1 JNDI程序包	83
4.2.3 确定关系	66	5.4.2 命名	83
4.3 创建表和列	66	5.4.3 上下文	83
4.3.1 选择数据类型	67	5.4.4 命名空间	84
4.3.2 创建关键字	68	5.5 JNDI操作	85
4.3.3 完成数据库物理设计	68	5.6 轻型目录访问协议	85
4.3.4 估计数据库大小	69	5.7 JNDI/LDAP实用范例	86
4.3.5 建立数据库开发环境	69	5.8 小结	88
4.4 定义应用程序	70	第6章 建立数据访问: JDBC	89
		6.1 介绍	89

6.1.1 什么是JDBC	90	6.14 范例研究: SilverStream应用服务器	115
6.1.2 JDBC的好处	90	6.14.1 SilverStream的数据库连接	115
6.2 JDBC结构: API和驱动	91	6.14.2 SilverStream内部的JDBC	116
6.3 JDBC API	93	6.14.3 从SilverStream数据库中的会话bean 访问JDBC	117
6.4 获得和更新数据	97	6.14.4 从会话bean访问JDBC	118
6.4.1 语句	97	6.14.5 在会话bean中定义资源引用查找	118
6.4.2 ResultSet	100	6.14.6 建立JDBC调用	118
6.5 SQL-to-Java 数据类型	101	6.14.7 建立连接	119
6.5.1 处理Null	102	6.14.8 建立、编译和执行SQL Select 语句	119
6.5.2 存储过程	102	6.14.9 创建环境入口	120
6.6 JDBC异常类型	103	6.14.10 把资源引用映射为SilverStream 数据库	120
6.6.1 SQLException	103	6.15 小结	121
6.6.2 SQLWarning	104	第7章 建立控制流: servlet	122
6.6.3 DataTruncation警告	104	7.1 什么是servlet	122
6.7 元数据	104	7.1.1 第一代——CGI	123
6.7.1 ResultSetMetaData对象	104	7.1.2 第二代——ISAPI/NSAPI	123
6.7.2 DatabaseMetaData接口	105	7.1.3 Java替代品——servlet	124
6.8 可滚动的结果集	105	7.2 servlet的优点	124
6.8.1 可滚动结果集方法	106	7.3 在MVC和应用范例中用做控制器	124
6.8.2 可滚动结果集范例	106	7.4 基本HTTP	125
6.8.3 可更新结果集	107	7.5 Servlet容器	127
6.8.4 创建可更新结果集	107	7.6 Servlet API	127
6.9 更新行	107	7.7 服务方法细节	128
6.9.1 删除行	108	7.7.1 把数据传递给客户端	128
6.9.2 插入行	108	7.7.2 设置MIME类型	129
6.10 事务支持	109	7.8 HTML客户	130
6.11 批处理语句	109	7.8.1 表单	130
6.12 JDBC 2.1新数据类型	111	7.8.2 经常使用的<FORM>标签属性	130
6.12.1 BLOB和CLOB	111	7.9 servlet生命周期	132
6.12.2 ARRAY	112	7.9.1 初始化	133
6.12.3 STRUCT	113	7.9.2 服务	133
6.12.4 REF	113	7.9.3 销毁	133
6.13 JDBC 2.0可选程序包API: javax.sql	113	7.10 ServletContext	134
6.13.1 使用JNDI进行数据库访问	113	7.11 HTTP请求报头	134
6.13.2 连接池	114		
6.13.3 分布式事务	115		
6.13.4 JDBC RowSets	115		

7.11.1 参数	134	8.3.2 JSP 1.1页面范例	166
7.11.2 内容	134	8.4 JSP结构	169
7.11.3 连接	135	8.4.1 简单的应用	169
7.11.4 Cookie	137	8.4.2 成熟的结构: 重定向请求	172
7.11.5 报头	137	8.4.3 模型-视图-控制器设计	175
7.12 HTTP响应报头	140	8.5 结论: JSP页面是企业Java解决方案中很	
7.13 会话管理	141	重要的组成部分	179
7.13.1 隐藏表单字段	141	8.5.1 JSP页面: 超越servlet的优势	180
7.13.2 改写URL	142	8.5.2 处理现有的servlet	180
7.13.3 持久Cookie	142	第9章 Java服务器页面: 使用标签库进行	
7.13.4 会话跟踪API	144	实际开发	181
7.14 发送请求	145	9.1 JSP语法	181
7.15 使用JDBC的servlet	146	9.1.1 JSP注释	181
7.16 Web应用	149	9.1.2 HTML注释	181
7.16.1 Web存档	149	9.1.3 隐藏注释	182
7.16.2 目录结构	149	9.1.4 JSP指令	182
7.16.3 部署描述符/映射到servlet的请求	150	9.1.5 JSP脚本元素	187
7.16.4 Multi-MIME类型	152	9.1.6 JSP动作	188
7.17 在SilverStream应用服务器中使用		9.1.7 对象作用域	198
servlet	152	9.1.8 隐含对象	198
7.18 Servlet 2.3 API	157	9.1.9 错误处理	199
7.19 小结	157	9.2 标签库	199
第8章 Java服务器页面: 介绍	158	9.2.1 声明标签库	199
8.1 JSP页面的特性	160	9.2.2 标签处理器	200
8.1.1 服务器端Java资源的高效利用	160	9.2.3 创建公司横幅定制标签	202
8.1.2 Web开发、部署和维护	160	9.3 小结	204
8.1.3 可重用的组件	161	第10章 为应用构建用户界面	205
8.1.4 分离业务逻辑和表示	161	10.1 模型-视图-控制器示例	205
8.1.5 大型开发团体和广泛的支持	161	10.1.1 优势	206
8.1.6 平台无关性	161	10.1.2 缺点	207
8.1.7 使用标签的下一代页面开发	161	10.2 Struts框架	207
8.2 JSP页面的各个组成部分	161	10.2.1 安装Struts	207
8.2.1 JSP指令	162	10.2.2 必备软件	208
8.2.2 JSP标签	162	10.2.3 Struts框架概括	208
8.2.3 脚本元素	163	10.2.4 JavaBeans和作用域	213
8.3 开发和部署JSP页面	163	10.2.5 构建模型组件	214
8.3.1 编写JSP页面	163	10.2.6 构建视图组件	215

10.2.7 构建控制器组件	223	11.12 事务和事务管理	248
10.2.8 访问关系数据库	230	11.12.1 分布式事务	248
10.3 小结	233	11.12.2 多事务	249
第11章 企业JavaBeans: 介绍	234	11.12.3 Java事务服务	249
11.1 企业JavaBeans概述	234	11.12.4 Java事务API	249
11.2 分布式程序设计概括	236	11.12.5 实体bean方法和事务属性	249
11.3 EJB框架	236	11.12.6 会话bean方法和事务属性	249
11.4 会话和实体bean	238	11.13 安全性	250
11.4.1 会话bean	238	11.14 部署	250
11.4.2 实体bean	238	11.14.1 部署描述符	250
11.4.3 使用会话bean封装实体bean	239	11.14.2 程序包层次图	250
11.4.4 bean相关性	239	11.15 EJB开发中的个人角色	251
11.5 bean的属性	239	11.15.1 实体bean开发人员	251
11.5.1 无状态bean	239	11.15.2 bean编写人员	251
11.5.2 有状态bean	240	11.15.3 应用组装人员	252
11.6 bean的各个组成部分	240	11.15.4 bean部署人员	252
11.6.1 Home接口	240	11.16 合作角色	252
11.6.2 EJBObject接口	240	11.16.1 组件提供者	252
11.6.3 部署描述符	240	11.16.2 应用服务器提供者	252
11.6.4 SessionContext和EntityContext 对象	241	11.16.3 EJB容器提供者	253
11.6.5 相关和细粒度对象	241	11.17 小结	253
11.7 容器管理持久性和bean管理持久性	241	第12章 企业JavaBeans: 构建会话bean	254
11.7.1 容器管理持久性	241	12.1 创建会话bean	254
11.7.2 bean管理持久性	242	12.1.1 无状态和有状态会话bean	254
11.8 企业JavaBeans的生命周期	242	12.1.2 定义会话bean类	255
11.9 Java消息服务和消息驱动bean	243	12.1.3 会话bean和继承	255
11.9.1 Java消息服务	243	12.1.4 会话bean接口	255
11.9.2 消息驱动bean	244	12.1.5 会话bean类方法	260
11.10 分布式程序设计服务	246	12.1.6 范例会话bean组件代码	261
11.10.1 命名和注册	246	12.1.7 存根、层次和对象范例代码	267
11.10.2 远程方法调用	246	12.2 打包和部署会话bean	294
11.10.3 协议	246	12.2.1 JAR内容	294
11.11 公共对象请求代理体系结构和远程 方法调用	246	12.2.2 部署	296
11.11.1 CORBA	247	12.3 修改bean代码	296
11.11.2 Java/RMI	247	12.4 从客户端查找会话bean	296
		12.4.1 InitialContext	296
		12.4.2 类型限制	297

12.4.3 查找对象和接口: Java命名和目录 接口	297	13.5.9 本地接口	320
12.4.4 使用EJBHome创建实例	298	13.6 EJB客户端	321
12.5 从客户端调用会话bean	298	13.6.1 远程客户端	321
12.5.1 为调用EJB编写客户端	298	13.6.2 本地客户端	322
12.5.2 不同类型的客户端	298	13.7 使用GUI工具配置、打包和部署实体 bean	322
12.6 小结	300	13.7.1 集成、设置和打包实体bean	322
第13章 企业JavaBeans: 构建实体bean ...	301	13.7.2 部署实体bean Jar文件	326
13.1 定义实体bean	301	13.8 小结	326
13.1.1 实体bean的用途	302	第14章 使用电子邮件传递信息:	
13.1.2 实体bean生命周期状态	302	JavaMail	327
13.2 容器和bean管理持久性	303	14.1 一般的电子邮件通信	327
13.2.1 bean管理持久性	304	14.2 JavaMail的任务	328
13.2.2 容器管理持久性	304	14.3 提供者注册表	329
13.3 CMP实体bean的剖析	305	14.4 JavaMail结构和主要的类	330
13.3.1 bean编写人员创建的组件	305	14.4.1 Java活动框架	331
13.3.2 容器创建的对象	307	14.4.2 javax.mail.Session	331
13.4 开发实体bean	308	14.4.3 javax.mail.Store	332
13.4.1 第一步——向数据库安装数据源 ...	308	14.4.4 javax.mail.Folder	333
13.4.2 第二步——开发主关键字类	309	14.4.5 javax.mail.search	334
13.4.3 第三步——开发实体bean类	309	14.4.6 javax.mail.Address	336
13.4.4 第四步——定义主接口或本地主 接口	312	14.4.7 javax.mail.internet.InternetAddress ...	336
13.4.5 第五步——定义本地或远程接口 ...	312	14.4.8 javax.mail.Message	336
13.4.6 第六步——定义部署描述符	313	14.4.9 javax.mail.internet.MimeMessage ...	337
13.4.7 第七步——使用容器提供的工具 进行部署	314	14.4.10 javax.mail.Transport	341
13.4.8 第八步——创建客户端应用程序 ...	314	14.5 发送和接收消息	341
13.5 深入了解实体bean的开发	314	14.5.1 发送因特网消息	341
13.5.1 主关键字	315	14.5.2 接收因特网消息	342
13.5.2 映射到单独的字段	315	14.5.3 处理多部分消息	343
13.5.3 EntityBean类和生命周期事件方法 ...	316	14.5.4 发送MimeMultipart消息	343
13.5.4 主接口和create()方法	318	14.5.5 接收MimeMultipart消息	344
13.5.5 主接口和finder()方法	319	14.5.6 发送消息中的二进制数据	345
13.5.6 主接口和remove()方法	320	14.5.7 接收消息中的二进制数据	347
13.5.7 主接口和getEJBMetaData()方法	320	14.6 在Java服务器页面中使用JavaMail ...	348
13.5.8 远程接口	320	14.6.1 简要指令	349
		14.6.2 详细指令	352
		14.6.3 代码解决方案	357

14.7 小结	359	16.2.2 实例部分	394
第15章 应用间的消息通信: Java消息 服务	360	16.2.3 元素	394
15.1 消息通信基础	360	16.2.4 标签	395
15.1.1 消息	360	16.2.5 空元素	395
15.1.2 中间件	360	16.2.6 属性	395
15.1.3 面向消息的中间件	361	16.2.7 实体	396
15.1.4 通信模式	361	16.2.8 转义字符	397
15.2 JMS的位置	362	16.3 未解析的数据	398
15.2.1 应用中的JMS	362	16.3.1 注释	398
15.2.2 JMS中的消息模型	363	16.3.2 CDATA	398
15.2.3 处理异常	364	16.4 处理指令	399
15.2.4 会话管理	365	16.4.1 字符引用	399
15.2.5 消息	365	16.4.2 字符范围	399
15.3 JMS组件	367	16.4.3 空格	399
15.3.1 管理对象	367	16.5 文档类型定义	400
15.3.2 接口类	367	16.6 XML模式	403
15.4 生成和销毁消息	368	16.6.1 XML模式与DTD的比较	403
15.4.1 MessageProducer	368	16.6.2 模式约束	404
15.4.2 MessageConsumer	368	16.6.3 XML模式的特性	404
15.4.3 MessageListener	369	16.6.4 模式词汇	405
15.4.4 MessageSelector	369	16.6.5 模式范例	405
15.5 JMS范例	369	16.6.6 模式的未来	410
15.5.1 简单的点对点范例	369	16.7 XML解析器	410
15.5.2 简单的发布/订阅范例	372	16.8 文档对象模型	411
15.5.3 点对点浏览范例	375	16.8.1 DOM结构模型	411
15.5.4 MessageListener范例	378	16.8.2 内存管理	412
15.5.5 MessageSelector范例	382	16.8.3 命名规范	413
15.5.6 队列和主题目的地维护范例	386	16.8.4 DOM对象	413
15.6 JMS实现和部署问题	390	16.9 XML简单API	413
15.6.1 还有其他什么需要实现的部分	390	16.10 XML输出	416
15.6.2 部署	390	16.11 XSL	417
15.7 小结	391	16.11.1 XSL转换	417
第16章 使用XML进行数据交换	392	16.11.2 XSL格式对象	418
16.1 什么是XML	392	16.11.3 XML路径语言	418
16.2 XML文档的结构	394	16.11.4 XSLT命名空间	418
16.2.1 序言	394	16.12 样式表结构	419
		16.12.1 stylesheet元素	419

16.12.2 模板元素	422	17.6.14 并行处理	437
16.13 应用样式表	422	17.6.15 测试覆盖分析	437
16.14 XML链接语言	422	17.7 Web站点测试工具和站点管理工具	437
16.15 XML安全	424	17.7.1 负载和性能测试工具	438
16.15.1 XML关键管理规范	424	17.7.2 Java测试工具	445
16.15.2 安全评估标记语言	424	17.7.3 链接检查工具	445
16.15.3 XKMS和SAML的关系	424	17.7.4 HTML确认器	446
16.16 小结	425	17.7.5 Web功能/衰减测试工具	447
第17章 验证应用	426	17.7.6 Web站点安全性测试工具	448
17.1 Java和测试	426	17.7.7 外部站点监控服务	449
17.2 用调试进行质量控制	428	17.7.8 Web站点管理工具	449
17.3 调试技术	429	17.7.9 其他Web测试工具	450
17.3.1 不靠主观判断进行测试	429	17.8 小结	450
17.3.2 散散心, 然后重新部署	429	第18章 执行应用	452
17.3.3 如何使调试工作更加科学	429	18.1 概述	452
17.3.4 语法错误	432	18.2 编写高性能应用	453
17.4 校正错误	432	18.2.1 规模因素	453
17.4.1 改善可靠性	433	18.2.2 性能规模	453
17.4.2 优秀调试器的重要性	434	18.2.3 组件设计	454
17.5 测试和开发阶段	434	18.2.4 环境因素	455
17.5.1 Alpha阶段	435	18.2.5 缩小网络流量	455
17.5.2 Beta阶段	435	18.2.6 编码指南	455
17.5.3 FCS	435	18.2.7 池管理	456
17.6 测试方法和技术	436	18.2.8 J2EE组件性能	459
17.6.1 单元测试	436	18.2.9 设计和性能	460
17.6.2 压力测试和海量数据生成	436	18.3 调整性能的准备工作的	461
17.6.3 代码复杂性分析	436	18.3.1 范围、目标和方法	461
17.6.4 基准/性能测试	436	18.3.2 负载测试	461
17.6.5 稳定性测试	436	18.3.3 Profiler	462
17.6.6 衰减测试	436	18.4 诊断和解决问题指南	463
17.6.7 黑盒测试	436	18.4.1 容易解决的问题	463
17.6.8 白盒测试	436	18.4.2 配置策略	463
17.6.9 可用性测试	436	18.4.3 改善applet下载速度	464
17.6.10 跨平台测试	437	18.4.4 把图像打包成一个类文件	464
17.6.11 安全性测试	437	18.4.5 使用JAR文件	464
17.6.12 安装测试	437	18.4.6 线程池	464
17.6.13 中断测试	437	18.5 影响服务器性能的因素	465

18.5.1 中央处理单元	465	19.2 组装过程	469
18.5.2 易失存储 (内存)	465	19.2.1 组装概述	469
18.5.3 永久存储 (磁盘)	465	19.2.2 组装应用和组件	470
18.5.4 数据库性能	465	19.3 部署过程	482
18.6 数据库监控和调整工具	467	19.3.1 使用J2EE引用实现进行部署	482
18.6.1 ShowPlan (SQL Server) 和Explain Plan (Oracle)	467	19.3.2 使用SilverStream应用服务器进行 部署	486
18.6.2 INDEX调整向导	467	19.3.3 使用BEA WebLogic服务器进行 部署	491
18.6.3 Oracle调整包	467	19.4 小结	518
18.6.4 Oracle SQL_TRACE	467	附录A 示例应用的文档	519
18.6.5 Optimizeit	467	附录B 相关工具	531
18.7 小结	467	附录C 快速参考资料	564
第19章 部署应用	469		
19.1 Java的“编写一次，随处运行”承诺	469		

第1章 Java、分布式计算和J2EE

本章包含如下内容：

- Java背景知识。
- 公司必须为网络经济进行改造。
- Web结构的需求。
- Web应用的生存周期。
- J2EE API和认证。
- XML和J2EE。
- J2EE应用的打包。
- 本书研究的应用范例。

在公司信息技术世界中，经常会出现新的发展范型：

- 在20世纪60年代，IBM为我们带来了叫做OS MVT/MFT的多任务操作系统。企业第一次可以在一台单独的机器上运行多个批处理任务。这预言着我们亲切地称之为“batch night cycle”的开始。无论商业上发生了什么事情，例如佣金/交易（brokerage/trades），整个公司的全部事务都可以在当天及时收集，并用键盘输入到穿孔卡片上。然后把这些信息转换成一个或多个批处理的COBOL任务，这些任务会记录相应的信息，创建公司自己的“工作簿和档案”。这项技术非常好，但信息总是在一天之后就过时了。
- 在20世纪70年代，IBM为我们带来了在线数据记录。称做CICS和VSAM的软件使这项功能成为可能。CICS提供了终端访问方式和数据入口。VSAM使用索引和关键字存储数据，便于数据的访问。这项技术非常好，而且现在的信息都相当及时，甚至当天的信息也能够非常及时。
- 在20世纪80年代，Microsoft改进了IBM所谓的“绿色屏幕”计算机，给我们带来了个人电脑，个人电脑配有鼠标和能够在本地存储数据的个人驱动空间。另外，其他供应商（包括IBM）为我们带来了SQL。因为SQL是由委员会制定，所以它变成了处理数据和数据库的事实上的标准。
- 在20世纪90年代，Microsoft普及了客户服务器平台。这是个很好的主意，它可谓提供了“用户友好”工作方式的典范，把商业事务同计算机联系在一起。但这里存在分布方面的问题。如果一个组织拥有1000台工作站，将不得不在软件的相同层上维护这些工作站，这是很困难的一件事情。
- 在2000年，Sun Microsystems（即：SUNW）和其他供应商为我们带来了Java 2 企业版（J2EE）。同样，委员会再次创建了构建商业处理的标准方式，该商业处理可以运行在大多数平台上。由于这些计算机类是可移植和可以相互协调工作的，所以功能非常强大。从开发的角度看，最主要的革新是SQL和J2EE，因为它们都是几乎每个人都拥护的标准。