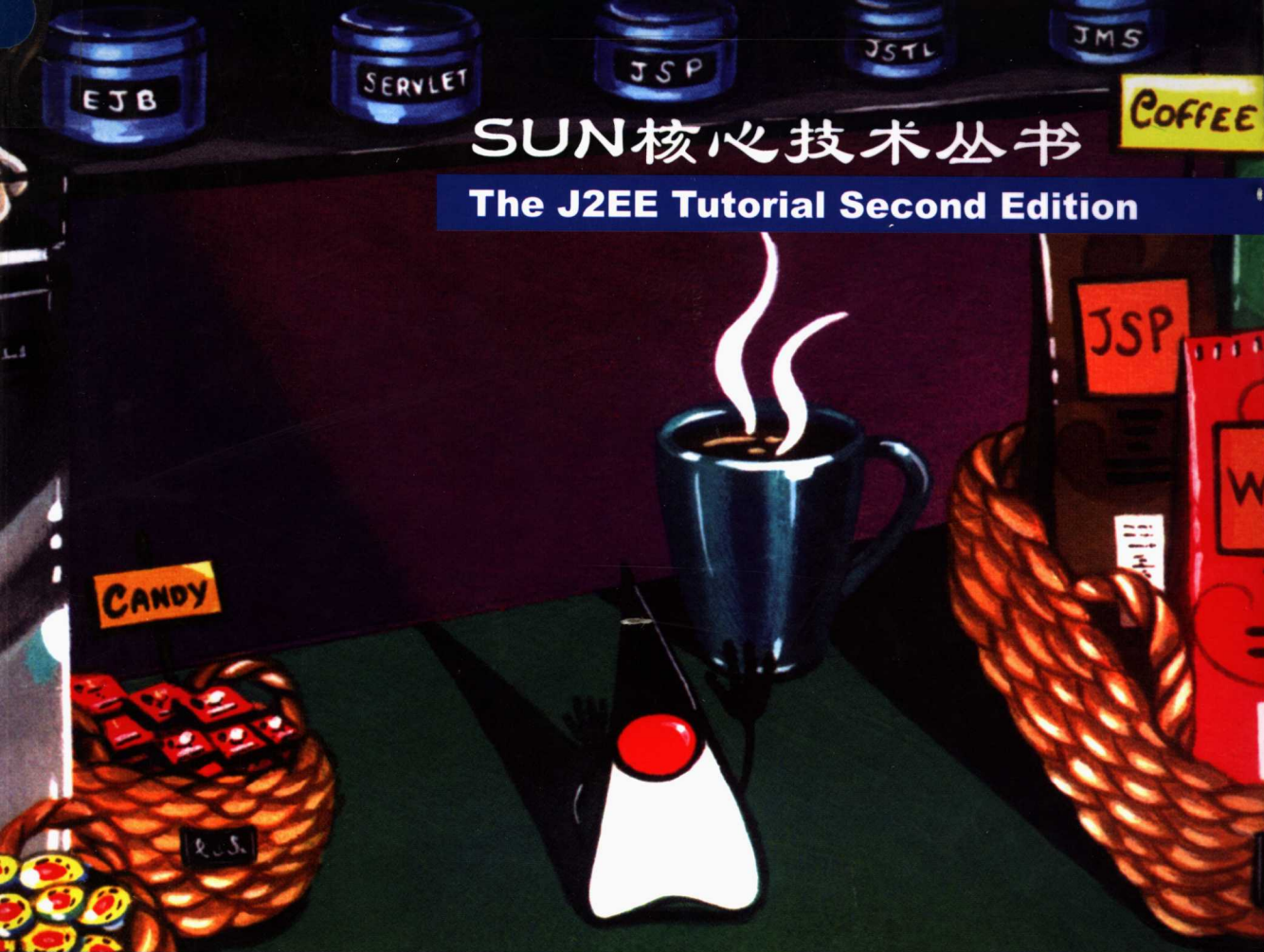




SUN核心技术丛书

The J2EE Tutorial Second Edition

J2EE 1.4

标准教材

(第二版)

[美] Stephanie Bodoff 等著

田玉敏 沈金河 等译

J2EE平台和应用服务器部门
副总裁Jeff Jackson作序

Addison
Wesley

 **Sun**
microsystems

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>


Java
Sun Microsystems

The J2EE Tutorial Second Edition

J2EE 1.4标准教材

(第二版)

〔美〕 Stephanie Bodoff 等著

田玉敏 沈金河 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本教材由7部分组成。第1部分讨论J2EE 1.4平台体系结构以及XML和Web应用程序的基本知识。第2部分讨论处理XML文档和实现Web服务组件的一些技术: JAXP、JAX-RPC、SAAJ和JAXR。第3部分讨论在开发J2EE或独立Web应用程序表示层时所使用的组件。第4部分讨论开发J2EE应用程序业务逻辑时使用的组件: 会话bean、实体bean、消息驱动bean和企业JavaBeans查询语言。第5部分讨论所有J2EE组件技术使用的系统服务: 事务、资源连接、安全连接和Java消息服务。第6部分给出一些案例分析, 它紧密围绕本教材所讨论的几种技术。最后一部分是附录, 包括Java编码方案、XML标准、HTTP概述、J2EE连接器体系结构。

本书适合于学习开发和部署J2EE 1.4应用程序的程序员阅读。



Authorized translation from the English language edition, entitled J2EE TUTORIAL, THE, 2nd Edition, 032124575X by BODOFF, STEPHANIE; GREEN, DALE; HAASE, KIM; JENDROCK, ERIC, published by Pearson Education, Inc, publishing as Addison Wesley Professional, Copyright©2004 by Sun Microsystems.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. CHINESE SIMPLIFIED language edition published by BEIJING MEDIA ELECTRONIC INFORMATION CO. LTD & PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, Copyright©2005

本书英文版由Pearson教育公司出版, Pearson教育公司已将简体中文独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可, 不得以任何形式和手段复制或抄袭本书内容。

版权贸易合同登记号 图字: 01-2004-5794

图书在版编目(CIP)数据

J2EE 1.4标准教材(第二版)/(美)博德欧夫(Bodoff, S.)等著; 田玉敏等译. —北京: 电子工业出版社, 2005.3

书名原文: The J2EE Tutorial Second Edition

ISBN 7-121-00943-9

I. J… II. ①博… ②田… III. JAVA语言—程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字(2005)第011064号

责任编辑: 李莹

印刷: 北京天竺颖华印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编: 100036

北京市海淀区翠微东里甲2号 邮编: 100036

经销: 各地新华书店

开本: 787×1092 1/16 印张: 65.75 字数: 1680千字

印次: 2005年3月第1次印刷

定价: 100.00元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换, 若书店售缺, 请与本社发行部联系。联系电话: 010-68279077。质量投诉请发邮件至zltts@phei.com.cn, 盗版侵权举报请发邮件至dbqq@phei.com.cn。

序

当《J2EE标准教材》的第一版发行时, Java 2 Platform, Enterprise Edition (J2EE) 在市场上还是新生儿。J2EE平台建立在J2SE (Java 2 Platform, Standard Edition) 的基础上, 它为企业应用服务器带来了“一次编写, 到处运行”API兼容性的好处。J2EE平台现在是1.4版本, 并且在应用服务器市场中具有广泛的一致性, 所以它已经牢固地奠定了作为企业应用服务器的标准的地位。

《J2EE 1.4标准教材(第二版)》论及了J2EE 1.4平台及更多内容。如果你阅读过本教材的第一版, 可能会注意到第二版本内容增到了三倍。这反映了J2EE平台的主要扩展和Sun Java System Application Server Platform Edition 8 (本教材所使用的软件) 中两种未来J2EE技术的可用性。

J2EE 1.4平台的一个最重要的新增功能是它本质上用JAX-RPC 1.1 API支持Web服务, 这使Web服务端点可以基于Servlet和企业bean。该平台还包含Web服务支持API, 这些API用于直接处理XML数据流(SAAJ) 和用于访问Web服务注册中心(JAXR)。此外, J2EE 1.4平台要求WS-I Basic Profile 1.0。这表明除了平台独立性和完全Web服务支持外, J2EE 1.4平台还提供Web服务互操作性。

J2EE 1.4平台包含对Java Servlet和JavaServer Pages (JSP) 技术的主要增强功能, 这些增强功能是Web层的基础。本教材还展示了两种激动人心的新技术: JavaServer Pages标准标签库(JSTL) 和JavaServer Faces, 这两种技术对于J2EE 1.4平台来说不是必需的, 但它们简化了构建J2EE应用程序用户界面的任务。这些新技术可以在Sun Java System Application Server中使用。不久以后, 它们将在新开发工具中提供, 并且下一版本的J2EE平台中非常可能包含这些技术。

熟悉核心J2EE平台企业bean技术的读者将会注意到一些主要的升级: 增加了以前提及的Web服务端点和计时器服务, 以及EJB QL和消息驱动bean的增强功能。

有了这些新特性, 我相信你会发现花时间和精力来采用J2EE 1.4平台是值得的。你可以增加开发J2EE应用程序的范围, 并且你的应用程序将会在最大可能范围内的应用服务器产品上运行。

为帮助你学习所有关于J2EE 1.4平台的知识, 《J2EE 1.4标准教材(第二版)》遵循了大家熟悉的Java系列教材的模型, 简洁描述了每种技术的本质特性, 并附带了许多可在Sun Java System Application Server上部署和运行的代码示例。阅读完本教材, 你将会成为下一波J2EE应用程序开发人员中的佼佼者。

Jeff Jackson

J2EE平台和应用服务器部门副总裁

Sun Microsystems

美国加利福尼亚州圣克拉拉市

目 录

第1章 概述	1
1.1 分布式多层应用程序	1
1.2 J2EE容器	5
1.3 Web服务支持	7
1.4 打包应用程序	8
1.5 开发角色	9
1.6 J2EE 1.4平台API	11
1.7 Sun Java System Application Server Platform Edition 8	15
第2章 了解XML	21
2.1 XML简介	21
2.2 生成XML数据	28
2.3 设计XML数据结构	53
2.4 小结	56
第3章 Web应用程序入门	57
3.1 Web应用程序生命周期	58
3.2 Web模块	60
3.3 配置Web应用程序	67
3.4 杜克Bookstore示例	70
3.5 从Web应用程序访问数据库	70
3.6 更多信息	73
第4章 用于XML处理的Java API	74
4.1 JAXP API	74
4.2 包概述	74
4.3 用于XML API的简单API	75
4.4 文档对象模型API	77
4.5 可扩展样式表语言转换API	78
4.6 使用JAXP库	79
4.7 后续	79
第5章 用于XML的简单API	81
5.1 何时使用SAX	81
5.2 回显具有SAX解析器的XML文件	82
5.3 添加其他事件处理程序	97

5.4	用非验证解析器处理错误	100
5.5	显示特殊字符和CDATA	106
5.6	用DTD解析	108
5.7	选择自己的解析器实现	113
5.8	使用验证解析器	113
5.9	解析参数化DTD	118
5.10	处理词法事件	119
5.11	使用DTDHandler和EntityResolver	125
5.12	更多信息	127
第6章	文档对象模型	128
6.1	何时使用DOM	128
6.2	将XML读到DOM中	133
6.3	显示DOM层次	138
6.4	检查DOM的结构	152
6.5	从DOM构建用户友好JTree	158
6.6	创建和操纵DOM	171
6.7	用XML架构验证	179
6.8	更多信息	183
第7章	可扩展样式表语言转换	184
7.1	XSL、XSLT和XPath简介	184
7.2	XPath如何工作	185
7.3	把DOM作为XML文件写出	192
7.4	从任意数据结构生成XML	198
7.5	用XSLT转换XML数据	210
7.6	用Xalan从命令行转换	229
7.7	用过滤器链连接转换	230
7.8	更多信息	236
第8章	利用JAX-RPC构建Web服务	237
8.1	设置端口	237
8.2	用JAX-RPC创建简单的Web服务和客户端	238
8.3	JAX-RPC支持的类型	245
8.4	Web服务客户端	246
8.5	Web服务互操作性和JAX-RPC	255
8.6	更多信息	256
第9章	SAAJ	257
9.1	SAAJ概述	257
9.2	教程	261

9.3 代码示例	280
9.4 更多信息	295
第10章 用于XML注册的Java API	296
10.1 JAXR概述	296
10.2 实现JAXR客户端	298
10.3 运行客户端示例	317
10.4 在J2EE应用程序中使用JAXR客户端	324
10.5 更多信息	330
第11章 Java Servlet技术	331
11.1 什么是servlet	331
11.2 示例servlet	331
11.3 servlet生命周期	335
11.4 共享信息	337
11.5 初始化servlet	340
11.6 编写服务方法	341
11.7 过滤请求和响应	345
11.8 调用其他Web资源	350
11.9 访问Web上下文	352
11.10 维护客户端状态	353
11.11 终结servlet	355
11.12 更多信息	358
第12章 JavaServer Pages技术	359
12.1 什么是JSP页面	359
12.2 JSP页面示例	362
12.3 JSP页面的生命周期	366
12.4 创建静态内容	370
12.5 创建动态内容	370
12.6 表达式语言	371
12.7 JavaBeans组件	377
12.8 使用自定义标签	381
12.9 在JSP页面中重用内容	384
12.10 将控件传输到另一Web组件	385
12.11 包括小程序	386
12.12 设置JSP页面组的属性	388
12.13 更多信息	390

第13章	JavaServer Pages文档	391
13.1	示例JSP文档	391
13.2	创建JSP文档	395
13.3	确定JSP文档到容器	406
第14章	JavaServer Pages标准标签库	407
14.1	示例JSP页	407
14.2	使用JSTL	410
14.3	核心标签库	411
14.4	XML标签库	416
14.5	国际化标签库	420
14.6	SQL标签库	422
14.7	函数	425
14.8	更多信息	427
第15章	JSP页中的自定义标签	428
15.1	什么是自定义标签	428
15.2	示例JSP页	429
15.3	标签的类型	431
15.4	用标签文件封装可重用内容	435
15.5	标签库描述符	446
15.6	编写简单标签处理程序	453
第16章	JSP页中的脚本	469
16.1	示例JSP页	469
16.2	使用脚本	470
16.3	禁用脚本	471
16.4	声明	471
16.5	scriptlet	472
16.6	表达式	472
16.7	编写接受脚本元素的标签的程序	473
第17章	JavaServer Faces技术	480
17.1	JavaServer Faces技术的优点	481
17.2	什么是JavaServer Faces应用程序	481
17.3	框架角色	482
17.4	简单JavaServer Faces应用程序	482
17.5	用户界面组件模型	489
17.6	导航模型	495
17.7	受支持bean的管理	496
17.8	各个部分如何进行组合	498

17.9	JavaServer Faces页的生命周期	500
17.10	更多信息	505
第18章	在JSP页中使用JavaServer Faces技术	506
18.1	JavaServer Faces示例应用程序	506
18.2	设置页面	509
18.3	使用核心标签	511
18.4	使用HTML组件标签	512
18.5	使用本地化消息	529
18.6	使用标准转换器	531
18.7	在组件上注册监听器	534
18.8	使用标准验证程序	535
18.9	把组件值和实例绑定到外部数据源	537
18.10	引用受支持bean方法	540
18.11	使用自定义对象	543
第19章	使用JavaServer Faces技术进行开发	547
19.1	编写组件属性	547
19.2	执行本地化	556
19.3	创建自定义转换器	558
19.4	实现事件监听器	560
19.5	创建自定义验证程序	563
19.6	编写受支持bean方法	567
第20章	创建自定义UI组件	571
20.1	确定是否需要自定义组件或呈现程序	571
20.2	了解图像映射示例	573
20.3	创建自定义组件的步骤	578
20.4	创建组件标签处理程序	578
20.5	在标签库描述符中定义自定义组件标签	582
20.6	创建自定义组件类	583
20.7	委托呈现给呈现程序	589
20.8	处理自定义组件的事件	591
第21章	配置JavaServer Faces应用程序	593
21.1	应用程序配置资源文件	593
21.2	配置bean	594
21.3	注册消息	601
21.4	注册自定义验证程序	602
21.5	注册自定义转换器	602
21.6	配置导航规则	603

21.7	向呈现工具包注册自定义呈现程序	605
21.8	注册自定义组件	607
21.9	JavaServer Faces应用程序的基本需求	607
第22章	国际化和本地化Web应用程序	613
22.1	Java平台本地化类	613
22.2	提供本地化消息和标签	613
22.3	日期和数字格式化	615
22.4	字符集和编码	616
22.5	更多信息	618
第23章	企业bean	619
23.1	什么是企业bean	619
23.2	什么是会话bean	620
23.3	什么是实体bean	621
23.4	什么是消息驱动bean	624
23.5	用接口定义客户端访问	625
23.6	企业bean的内容	629
23.7	企业bean的命名约定	629
23.8	企业bean的生命周期	630
23.9	更多信息	633
第24章	企业bean入门	634
24.1	创建J2EE应用程序	634
24.2	创建企业bean	635
24.3	创建应用程序客户端	638
24.4	创建Web客户端	641
24.5	映射企业bean引用	644
24.6	指定Web客户端的上下文根	645
24.7	部署J2EE应用程序	645
24.8	运行应用程序客户端	646
24.9	运行Web客户端	646
24.10	修改J2EE应用程序	647
第25章	会话bean示例	649
25.1	CartBean示例	649
25.2	Web服务示例: HelloServiceBean	657
25.3	其他企业bean特性	661
25.4	使用计时器服务	663
25.5	处理异常	670

第26章 bean管理持久性示例	671
26.1 SavingsAccountBean示例	671
26.2 映射bean管理持久性的表关系	683
26.3 bean管理持久性的主键	694
26.4 针对具有bean管理持久性的实体bean的deploytool技巧	697
第27章 容器管理持久性示例	698
27.1 RosterApp应用程序概述	698
27.2 PlayBean的代码	699
27.3 RosterApp中的方法调用	703
27.4 编译和运行RosterApp示例	712
27.5 RosterApp设置指导旅行	723
27.6 容器管理持久性的主键	730
27.7 高级CMP主题: OrderApp示例	731
27.8 针对具有容器管理持久性的实体bean的deploytool技巧	744
第28章 消息驱动bean示例	746
28.1 示例应用程序概述	746
28.2 应用程序客户端	746
28.3 消息驱动bean类	747
28.4 部署和运行SimpleMessageApp	749
28.5 消息驱动bean的deploytool技巧	751
28.6 发送消息的组件的deploytool技巧	752
第29章 企业JavaBeans查询语言	754
29.1 术语	754
29.2 简化的语法	754
29.3 查询示例	755
29.4 完整语法	759
29.5 EJB QL限制	775
第30章 事务	776
30.1 什么是事务	776
30.2 容器管理事务	776
30.3 bean管理的事务	784
30.4 企业bean的事务选项小结	793
30.5 事务超时	793
30.6 隔离级别	794
30.7 更新多个数据库	794
30.8 Web组件中的事务	795

第31章 资源连接	796
31.1 JNDI命名	796
31.2 DataSource对象和连接池	797
31.3 数据库连接	797
31.4 邮件会话连接	800
31.5 URL连接	802
31.6 更多信息	804
第32章 安全	805
32.1 概述	805
32.2 域、用户、组和角色	805
32.3 Web层安全	808
32.4 了解登录身份验证	813
32.5 安装和配置SSL支持	823
32.6 XML和Web服务安全	831
32.7 EJB层安全	844
32.8 应用程序客户端层安全	846
32.9 EIS层安全	847
32.10 传播安全标识	849
32.11 什么是容器的Java授权联系人	850
32.12 更多信息	850
第33章 Java消息服务API	851
33.1 概述	851
33.2 基本的JMS API概念	853
33.3 JMS API编程模型	856
33.4 编写简单的JMS客户端应用程序	864
33.5 创建强健的JMS应用程序	879
33.6 在J2EE应用程序中使用JMS API	895
33.7 更多信息	899
第34章 使用JMS API的J2EE示例	900
34.1 把JMS API与会话Bean一起使用的J2EE应用程序	900
34.2 把JMS API与实体bean一起使用的J2EE应用程序	907
34.3 从远程J2EE服务器消费消息的应用程序示例	912
34.4 在两台J2EE服务器上部署消息驱动bean的应用程序示例	917
第35章 Coffee Break应用程序	924
35.1 公共代码	925
35.2 JAX-RPC咖啡供应商服务	925
35.3 SAAJ咖啡供应商服务	932

35.4	Coffee Break服务器	945
35.5	JavaServer Faces版本的Coffee Break服务器	948
35.6	编译、打包、部署和运行应用程序	953
第36章	杜克银行应用程序	960
36.1	企业bean	960
36.2	应用程序客户端	965
36.3	Web客户端	971
36.4	国际化	979
36.5	编译、打包、部署和运行应用程序	980
36.6	运行客户端	987
附录A	Java编码方案	989
附录B	XML及相关规范	991
附录C	HTTP概述	999
附录D	J2EE连接器体系结构	1000
术语表		1005
关于作者		1030

第1章 概 述

今天，越来越多的开发人员想为企业编写分布式事务型应用程序，并因此利用服务器端技术的速度、安全和可靠性。如果你已经涉足这个领域，那么你知道在电子交易和信息技术的流转迅速并且要求高的领域中，必须设计、构建并生产企业应用程序，以缩减所需资金、获得更高的速度并减少所需资源。

要缩减成本并快速跟踪应用程序设计和开发，可使用Java™ 2 Platform, Enterprise Edition (J2EE™)，它提供了一种基于组件的方法来设计、开发、组装和部署企业应用程序。J2EE平台提供了一个多层次的分布式应用模型、一些可重用的组件、一个统一的安全模型、灵活的事务控制，并借助基于可扩展标记语言 (XML) 的开放标准和协议的集成数据交换提供Web服务支持。

你不仅可以比以往更快地向市场提供创新的业务解决方案，而且独立于平台并且基于组件的J2EE解决方案不会受到任何供应商的产品和应用程序编程接口 (API) 的制约。供应商和客户都能自由地选择最适合他们的业务和技术需求的产品和组件。

本教材使用一些例子来描述Sun Java System Application Server Platform Edition 8中开发企业应用可以使用的特性和功能。不管你是新手还是经验丰富的开发人员，都会发现这些例子和相应的课文是创建自己的解决方案的一个有价值、方便的知识基础。

如果你是J2EE企业应用程序开发的新手，那么本章就是良好的起点。这里我们将回顾开发基础，学习J2EE体系结构和API，熟悉重要的术语和概念，并了解如何编程、组装和部署J2EE应用程序。

1.1 分布式多层应用程序

对于企业应用程序，J2EE平台使用分布式多层应用程序模型。根据功能将应用程序逻辑分成一些组件，而组成J2EE应用程序的各种应用组件根据该应用程序组件在多层次J2EE环境中所属的层次，将这些组件安装在不同的计算机上。图1-1展示了两个多层次J2EE应用，它们分成以下列表中描述的层次。图1-1中所描述的J2EE应用程序部分在第1.1.1节“J2EE组件”中介绍。

- 客户端层组件运行在客户端计算机上
- Web层组件运行在J2EE服务器上
- 业务层组件运行在J2EE服务器上
- 企业信息系统 (EIS) 层软件运行在EIS服务器上

尽管J2EE应用程序可以包含图1-1中所示的3或4个层次，但是人们通常认为J2EE多层次应用程序是3层应用程序，因为它们分布在3个位置：客户端计算机、J2EE服务器计算机和后台的数据库或遗留计算机。以这种方式运行的3层应用程序通过在客户端应用程序和后台存储之前放置一个多线程的应用服务器，从而扩展了标准的2层客户端和服务器模型。

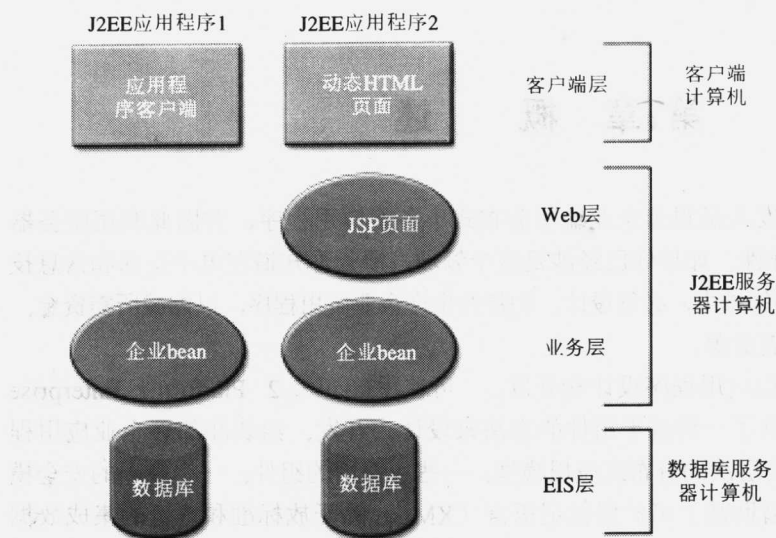


图1-1 多层应用程序

1.1.1 J2EE组件

J2EE应用程序是由组件组成的。J2EE组件是独立的功能性软件单元，它利用其相关的类和文件组装到J2EE应用中，并和其他组件进行通信。J2EE规范定义了以下J2EE组件：

- 应用程序客户端和applet是运行在客户端的组件
- Java Servlet和JavaServer Pages™ (JSP™) 技术组件是运行在服务器上的Web组件
- 企业JavaBeans™ (EJB™) 组件（企业bean）是运行在服务器上的业务组件

J2EE组件是用Java编程语言编写的，并且它的编译方式和该语言的任何程序相同。J2EE组件和“标准的”Java类之间的不同在于J2EE组件被组装到一个J2EE应用程序中，并被证实是格式良好的，和J2EE规范兼容，并且被部署到产品中（其中，J2EE服务器运行和管理这些组件）。

1.1.2 J2EE客户端

J2EE客户端可以是Web客户端或应用程序客户端。

Web客户端

Web客户端包括两个部分：（1）动态Web页面包含各种类型的标记语言（HTML、XML，等等），它们是由运行在Web层中的Web组件生成的；（2）一个Web浏览器，它呈现从服务器接收的页面。

Web客户端有时被称做瘦客户端。瘦客户端一般不查询数据库，执行复杂业务规则，也不连接到遗留应用程序。使用瘦客户端时，执行在J2EE服务器上的企业bean摆脱了这样的重量级操作，其中这些bean可以利用J2EE服务器端技术的安全性、速度、服务和可靠性。

applet

从Web层接收的Web页面可以包含嵌入式applet。applet是用Java编程语言编写的小型客户端应用程序，它被安装在Web浏览器中的Java虚拟机上执行。然而，客户端系统可能需要Java插件以及可能的安全策略文件，以便该applet成功地在Web浏览器中执行。

Web组件是创建Web客户端程序的首选API，因为客户端系统不需要插件或安全策略文件。并且，使用Web组件可以支持更简洁并且更模块化的应用程序设计，因为它们提供了一种从Web页面设计中分离应用程序编程的方法。Web页面设计所涉及的人员不必了解Java编程语言的语法就能完成他们的工作。

应用程序客户端

应用程序客户端在客户端计算机上运行，并且为用户提供了一种方法来处理以下任务：这些任务需要的用户界面比标记语言可以提供的界面更丰富。应用程序客户端通常具有一个用Swing或Abstract Window Toolkit (AWT) API创建的图形用户界面 (GUI)，但是命令行界面当然也可以。

应用程序客户端直接访问运行在业务层的企业bean。然而，如果应用需求需要，那么应用程序客户端可以打开一个HTTP连接，以便与运行在Web层的servlet建立通信。

JavaBeans™组件体系结构

服务器和客户端层可能还包括基于JavaBean组件体系结构的组件（JavaBeans组件），以管理应用程序客户端或applet与运行在J2EE服务器上的组件（或服务器组件和数据库）之间的数据流。J2EE规范并不将JavaBeans组件看做J2EE组件。

JavaBeans组件有一些属性，并且有访问这些属性的get和set方法。以这种方式使用的JavaBeans组件在设计和实现方面通常很简单，但是应该符合JavaBeans组件体系结构中的命名和设计约定。

J2EE服务器通信

图1-2展示了可以组成客户端层的各种元素。该客户端和运行在J2EE服务器上的业务层直接进行通信，或者，如果与运行在一个浏览器中的客户端进行通信，则通过JSP页面或运行在Web层中的servlet。

开发者的J2EE应用程序使用基于浏览器的瘦客户端或胖应用程序客户端。在决定使用哪种客户端时，应该考虑保持客户端的功能并接近用户（胖客户端）和尽可能多地将功能转移到服务器（瘦客户端）之间的得失。向服务器转移的功能越多，分布、部署和管理应用程序就越简单；然而，在客户端保持更多功能可以导致更容易感知的用户体验。

1.1.3 Web组件

J2EE Web组件是servlet或使用JSP技术创建的页面（JSP页面）。servlet是Java编程语言类，它动态地处理请求和构造响应。JSP页面是基于文本的文档，它作为servlet执行，但是允许使用一种更自然的方法来创建静态内容。

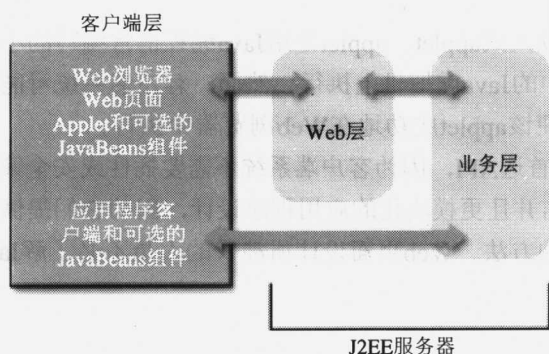


图1-2 服务器通信

在应用程序组装期间，静态HTML页面及applet和Web组件捆绑在一起，但是J2EE规范并不将其看做Web组件。服务器端工具类也可以和Web组件捆绑在一起，并且和HTML页面一样，它也不被看做Web组件。

如图1-3所示，Web层和客户端层一样，可能包含一个JavaBeans组件来管理用户输入并将该输入发送给运行在业务层中的企业bean以进行处理。

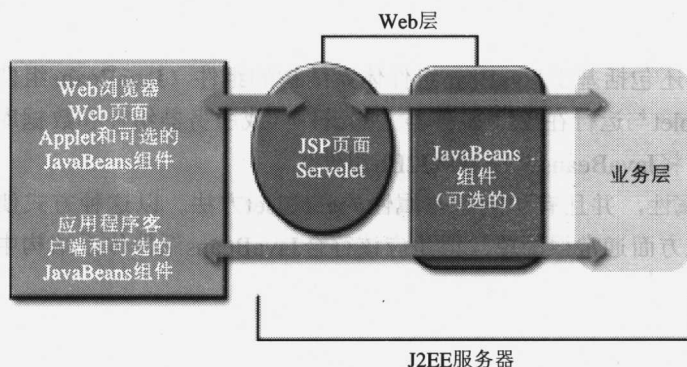


图1-3 Web层和J2EE应用程序

1.1.4 业务组件

业务代码，即解决或满足特定业务领域（如银行、零售或金融）需求的逻辑，是由运行在业务层中的企业bean处理的。图1-4展示了企业bean如何接收来自客户端程序的数据，并处理这些数据（如果需要的话），再将其发送到企业信息系统中以便存储。企业bean还从存储中检索数据，处理这些数据（如果需要的话），并且将其发送回客户端程序。

企业bean有3种类型：会话bean、实体bean和消息驱动bean。会话bean描述与客户的短暂交谈。客户完成执行时，会话bean及其数据就会消失。相反，实体bean描述存储在数据库表一行中的持久数据。如果客户终止或者服务器停机，那么底层的服务将确保保存了实体bean数据。消息驱动bean结合了会话bean和Java消息服务（JMS）消息监听器的特性，允许业务组件异步接收JMS消息。