

IBM大学合作项目书籍出版资助

J2EE

应用开发

实例精解

(WAS+RAD)

吴 镒 主编

黄鹏鹤 李东艳 崔崇芳 编著
冯瑞芳 段 竹 李光宇



清华大学出版社

IBM大学合作项目书籍出版资助

J2EE

应用开发 实例精解 (WAS+RAD)

吴 镒 主编

黄鹏鹤 李东艳 崔崇芳 编著
冯瑞芳 段 竹 李光宇

清华大学出版社

北京

内 容 简 介

本书共分为14章,包括两大部分:基础篇和综合应用篇。“基础篇”从环境部署、开发工具的安装讲起,由浅入深,详细讲述使用RAD、WAS和DB2产品进行J2EE项目开发的方法,同时结合每一章的知识点讲述相关实例,加深对知识点的理解。“综合应用篇”共讲述6个综合案例,每一个案例都严格按照软件工程的规范,详细讲述项目的背景与目标、需求分析、总体设计、数据库设计、详细设计与实现,使读者深刻体会项目开发的各个环节,提升综合开发能力和实际动手能力。案例中使用了传统的MVC开发方式和基于Struts框架的开发方式,通过对两种开发方式的学习,可以对J2EE开发技术有更加深入和透彻的理解。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

J2EE 应用开发实例精解(WAS+RAD)/吴镒主编. —北京:清华大学出版社,2011.3

ISBN 978-7-302-24117-1

I. ①J… II. ①吴… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 232271 号

责任编辑:梁 颖 高买花

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:三河市君旺印装厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:15.25 字 数:432千字

版 次:2011年3月第1版 印 次:2011年3月第1次印刷

印 数:1~4000

定 价:25.00元



FOREWORD

前言

J2EE(Java2 Enterprise Edition)是业界十分流行的应用程序开发架构,并已成为一个广泛认可的工业标准。作为 J2EE 技术的倡导者之一,IBM 公司凭借着卓越的市场洞察力和强大的研发能力,其 J2EE 应用开发、测试、调试和部署工具占据着庞大的市场份额。本书将以 IBM 公司的 RAD、WAS 和 DB2 为基础讲解 J2EE 应用开发技术。

RAD(Rational Application Developer)是 IBM 公司开发的基于 Eclipse 平台的面向开发、测试、调试和部署的集成开发平台; WAS(WebSphere Application Server)是业内广泛部署的 J2EE 和 Web 服务平台,是 IBM WebSphere 软件平台的基础,有着非常广泛的应用基础,目前占据着第一的市场份额。DB2 是 IBM 公司推出的一款十分流行的数据库产品,它支持从 PC 到 UNIX、从中小型机到大型机、从 IBM 到非 IBM(HP 及 SUN UNIX 系统等)的各种操作平台,具有很好的应用前景。据 IDC 统计,DB2 在全球数据库市场份额占第二位,仅次于 Oracle。

本书共分为 14 章,包括两大部分:基础篇和综合应用篇。“基础篇”由第 1 章到第 8 章组成,从环境部署、开发工具的安装讲起,由浅入深,详细讲述使用 RAD、WAS 和 DB2 产品进行 J2EE 项目开发的方法,同时结合每一章的知识点讲述相关实例,加深学生对知识点的理解。其中,第 1 章从概念上对 J2EE 进行了全面的总体描述,着重介绍了 J2EE 的基本概念、J2EE 平台的角色、J2EE 的体系架构以及 J2EE 的核心技术;第 2 章介绍了 IBM 公司的应用服务器 WAS 的基本特性,着重介绍了 WAS 的安装步骤以及如何部署企业级应用程序;第 3 章介绍了 IBM 公司的集成开发平台 RAD 的特性、RAD 的安装过程以及如何设置开发环境从而满足您的需要,同时介绍了 RAD 开发平台的基本操作界面;第 4 章介绍了 DB2 的基本概念、安装方法以及 JDBC 驱动的结构,着重介绍了在 RAD 中访问 DB2 数据库和创建数据库的基本方法;第 5 章介绍了 Servlet 的基本概念和编程接口,着重介绍了使用 RAD 进行 Servlet 开发、部署的基本方法和步骤,最后介绍了 Servlet 过滤器的开发;第 6 章介绍了 JSP 的基本概念、内置对象以及基本语法结构,着重介绍了在 RAD 中创建、运行 JSP 文件以及如何创建一个数据库并在 JSP 文件中显示的方法和步骤;第 7 章介绍了 JavaBean 的基本概念、结构、特性,以及在 JSP 中调用 JavaBean 的基本语法,讲解在 RAD 环境中如何通过 JSP 调用 JavaBean;第 8 章介绍了 Struts 架构的基本概念、流程以及核心 API,讲解如何在 RAD 中创建基于 Struts 架构的 Web 应用程序。

“综合应用篇”由第 9 章到第 14 章组成,讲述了 6 个综合案例,每一个案例我

们都严格按照软件工程的规范,详细讲述项目的背景与目标、需求分析、总体设计、数据库设计、详细设计与实现,使学生深刻体会项目开发的各个环节,提升综合开发能力和实际动手能力。案例中使用了传统的 MVC 开发方式和基于 Struts 框架的开发方式,通过对两种开发方式的学习,学生可以对 J2EE 开发技术有更加深入和透彻的理解。

本书可作为全国各类高等院校“J2EE 技术”相关课程的主讲教材、教学辅导教材、教师参考教材、课程设计教材以及毕业设计辅导教材,并配有多媒体课件,主要适合于本科生和大专(高职)学生。本书将会为那些想使用 J2EE 架构开发应用程序的初学者和中级读者提供有效的学习捷径。

由于作者水平有限,编写时间仓促,书中难免存在错误和不妥之处,敬请广大读者和专家批评指正。

编 者

2010 年 6 月



CONTENTS

目 录

基础篇

第 1 章 J2EE 概述	1
1.1 什么是 J2EE	1
1.2 J2EE 的优势	1
1.3 J2EE 的构成	2
1.4 J2EE 平台的角色	3
1.5 J2EE 体系架构	4
1.6 J2EE 核心技术	6
本章小结	8
习题 1	8
第 2 章 IBM WAS 的安装与配置	9
2.1 WAS 简介	9
2.2 WAS 的安装	9
2.3 部署企业级应用程序	13
本章小结	14
习题 2	14
第 3 章 IBM RAD 的安装与使用	15
3.1 RAD 概述	15
3.1.1 RAD 在 Rational 软件开发平台中的位置	15
3.1.2 RAD 6.0 支持的新特性	16
3.2 RAD 安装	17
3.3 RAD 工作台基础	21
3.3.1 工作台参数选择对话框	21
3.3.2 导入和导出	22
3.4 透视图	25
3.4.1 视图(View)	26
3.4.2 编辑器(Editor)	26

3.4.3 工作区(Workspace)管理	27
本章小结	28
习题 3	28
第 4 章 DB2 数据库的安装与使用	30
4.1 DB2 简介	30
4.2 DB2 的安装与管理	30
4.3 RAD 的数据访问功能	35
4.3.1 JDBC 概述	35
4.3.2 RAD 的数据库特性	36
4.3.3 RAD 的数据透视图	36
4.4 通过 RAD 访问 DB2 数据库	37
4.5 通过 RAD 创建数据库	40
4.5.1 创建数据库	40
4.5.2 创建模式	41
4.5.3 创建表	41
4.5.4 生成 DDL 文件	42
4.6 CloudScape	43
本章小结	44
习题 4	44
第 5 章 使用 RAD 进行 Servlet 开发	46
5.1 Servlet 概述	46
5.1.1 Servlet 起源	46
5.1.2 Servlet 与 Applet、CGI	46
5.1.3 Servlet 生命周期	47
5.1.4 在 RAD 中跟踪 Servlet 生命周期	49
5.2 Servlet 编程接口	55
5.3 使用部署描述符编辑器配置 Servlet	58
5.3.1 详细信息	59
5.3.2 URL 映射	59
5.3.3 初始化	59
5.3.4 启动时装入	60
5.3.5 安全角色引用	60
5.3.6 运行方式	61
5.3.7 图标	61
5.4 使用 RAD 开发 Servlet 实例	62
5.5 Servlet 过滤器开发	65
5.5.1 什么是 Servlet 过滤器	65
5.5.2 Servlet 过滤器工作原理	66

5.5.3	Servlet 过滤器的作用	66
5.5.4	Servlet 过滤器接口的构成	66
5.5.5	在 RAD 中创建 Servlet 过滤器	66
	本章小结	69
	习题 5	69
第 6 章	使用 RAD 进行 JSP 开发	71
6.1	JSP 概述	71
6.1.1	JSP 简介	71
6.1.2	JSP 2.0 介绍	72
6.1.3	JSP 技术的优势	72
6.1.4	JSP 内置对象	72
6.1.5	JSP 基本语法	76
6.2	RAD 中的 JSP 开发	80
6.3	JSP 高级话题	83
6.3.1	创建一个动态 Web 项目	83
6.3.2	创建一个 JSP 文件	84
6.3.3	在浏览器中运行 JSP 文件	85
6.3.4	创建一个数据库连接	86
6.3.5	创建一个数据库定义	87
6.3.6	创建一条插入语句	89
6.3.7	修改 JSP 文件以显示数据库内容	90
6.3.8	在浏览器中运行 JSP 文件	91
	本章小结	92
	习题 6	92
第 7 章	使用 RAD 进行 JavaBean 开发	94
7.1	什么是 JavaBean	94
7.2	JavaBean 的构成	94
7.2.1	JavaBean 属性	94
7.2.2	JavaBean 方法	96
7.2.3	JavaBean 事件	96
7.3	JavaBean 的特性	96
7.4	JSP 中调用 JavaBean	96
7.5	JavaBean 应用实例	97
	本章小结	100
	习题 7	100
第 8 章	Struts 框架	101
8.1	Struts 简介	101

8.2	Struts 流程	102
8.3	Struts 的核心 API	103
8.3.1	Struts 配置文件(Struts-config.xml)	103
8.3.2	ActionServlet 类	106
8.3.3	ActionMapping 类	107
8.3.4	Action 类	107
8.3.5	ActionForm 类	108
8.3.6	ActionForward 类	108
8.3.7	错误处理	108
8.4	RAD 中构建基于 Struts 架构的 Web 应用程序	110
8.4.1	RAD 中为基于 Struts 的 Web 应用程序提供的支持	110
8.4.2	构建基于 Struts 的 Web 应用程序	110
	本章小结	117
	习题 8	118

综合应用篇

第 9 章	大学生选课系统的设计与实现	119
9.1	项目背景与目标	119
9.2	需求分析	119
9.3	总体设计	120
9.4	数据库设计	121
9.4.1	数据库概念结构设计	121
9.4.2	数据库主要表结构说明	122
9.5	传统 MVC 模式的详细设计与实现	123
9.5.1	登录与注册模块的设计与实现	123
9.5.2	学生系统模块的设计与实现	128
9.5.3	教师系统的设计与实现	129
9.5.4	管理员模块的设计与实现	131
	本章小结	134
第 10 章	天海网上书城系统的设计与实现	135
10.1	项目背景与目标	135
10.2	需求分析	135
10.3	总体设计	137
10.4	数据库设计	137
10.4.1	数据库概念结构设计	137
10.4.2	数据库主要表结构说明	139
10.5	传统 MVC 模式的详细设计与实现	142
10.5.1	登录与注册模块的设计与实现	142

10.5.2 用户留言模块的设计与实现	144
10.5.3 购物车模块的设计与实现	146
10.5.4 后台管理模块的设计与实现	151
本章小结	154
第 11 章 晋亿员工个人信息管理系统的设计与实现	155
11.1 项目背景与目标	155
11.2 需求分析	155
11.2.1 可行性研究	155
11.2.2 需求分析	156
11.3 总体设计	156
11.3.1 设计思想	157
11.3.2 功能模块	157
11.4 数据库设计	158
11.4.1 数据库概念结构设计	158
11.4.2 数据库主要表结构说明	159
11.5 传统 MVC 模式的详细设计与实现	161
11.5.1 登录模块的设计与实现	161
11.5.2 机构管理模块的设计与实现	163
11.5.3 员工管理模块的设计与实现	164
11.5.4 个人信息管理模块的设计与实现	165
11.5.5 个人记事本管理模块的设计与实现	167
11.5.6 个人通讯录管理模块的设计与实现	168
本章小结	169
第 12 章 在线综合考试管理系统的设计与实现	170
12.1 项目背景与目标	170
12.2 需求分析	170
12.3 总体设计	171
12.4 数据库设计	173
12.4.1 数据库概念结构设计	173
12.4.2 数据库主要表结构说明	175
12.5 传统 MVC 模式的详细设计与实现	179
12.5.1 登录模块的设计与实现	179
12.5.2 管理员模块的设计与实现	180
12.5.3 教师管理模块的设计与实现	182
12.5.4 学生模块的设计与实现	184
12.5.5 关键技术研究与分析	185
本章小结	188

第 13 章 家乐超市管理系统的设计与实现	189
13.1 项目背景与目标	189
13.2 需求分析	189
13.3 总体设计	190
13.3.1 具体目标	190
13.3.2 系统功能模块图	191
13.4 数据库设计	191
13.4.1 数据库概念结构设计	191
13.4.2 数据库主要表结构说明	193
13.5 传统 MVC 模式的详细设计与实现	194
13.5.1 登录模块的设计与实现	194
13.5.2 人事管理模块的设计与实现	195
13.5.3 销售管理模块的设计与实现	198
13.5.4 进货管理模块的设计与实现	200
13.5.5 库存管理模块的设计与实现	203
本章小结	204
第 14 章 红旗图书馆图书管理系统的设计与开发	205
14.1 项目背景与目标	205
14.2 需求分析	205
14.3 总体设计	208
14.4 数据库设计	209
14.4.1 数据库概念结构设计	209
14.4.2 数据库主要表结构说明	211
14.5 Struts 架构的详细设计与实现	214
14.5.1 数据库与 Web 服务器连接的实现	214
14.5.2 登录模块的设计与实现	216
14.5.3 系统管理员模块的设计与实现	218
14.5.4 读者管理模块的设计与实现	220
14.5.5 系统查询模块的设计与实现	223
14.5.6 图书管理模块的设计与实现	224
14.5.7 图书借还模块的设计与实现	228
本章小结	230
附录 习题参考答案	231
参考文献	233

本章从概念上对 J2EE 进行全面的总体描述,着重介绍 J2EE 的基本概念、J2EE 平台的角色、J2EE 的体系架构以及 J2EE 的核心技术。

1.1 什么是 J2EE

J2EE 的全称是 Java2 企业版(Java2 Enterprise Edition),它是由 Sun Microsystems 公司提出并协同 IBM、Oracle 等厂商共同制定的一套全然不同于传统应用开发的技术架构,并已成为一个广泛认可的工业标准。J2EE 中包含许多组件,主要可简化且规范应用系统的开发与部署,进而提高可移植性、安全性与再用价值。

J2EE 核心是一组技术规范与指南,其中所包含的各类组件、服务架构及技术层次,均有共同的标准及规格,使各种依循 J2EE 架构的不同平台之间,存在良好的兼容性,解决过去企业后端使用的信息产品彼此之间无法兼容,导致企业内部或外部难以互通的窘境。

1.2 J2EE 的优势

J2EE 为搭建具有可伸缩性、灵活性、易维护性的商务系统提供了良好的机制。

1. 保留现存的 IT 资产

由于企业必须适应新的商业需求,利用已有的企业信息系统方面的投资,而不是重新制定全盘方案就变得很重要。这样,一个以渐进的(而不是激进的、全盘否定的)方式建立在已有系统之上的服务器端平台机制是公司所需求的。J2EE 架构可以充分利用用户原有的投资,如一些公司使用的 BEA Tuxedo、IBM CICS、IBM Encina、Inprise VisiBroker 以及 Netscape Application Server。这之所以成为可能是因为 J2EE 拥有广泛的业界支持和一些重要的“企业计算”领域供应商的参与。每一个供应商都对现有的客户提供了不用废弃已有投资,进入可移植的 J2EE 领域的升级途径。由于基于 J2EE 平台的产品几乎能够在任何操作系统和硬件配置上运行,所以现有的操作系统和硬件也能被保留使用。

2. 高效的开发

J2EE 允许公司把一些通用的、很烦琐的服务端任务交给中间件供应商去完成,这样开发人员可以集中精力在如何创建商业逻辑上,相应地缩短了开发时间。高级中间件供应商提供以下这些复杂的中间件服务:

1) 状态管理服务

让开发人员写更少的代码,不用关心如何管理状态,这样能够更快地完成程序开发。

2) 持续性服务

让开发人员不用对数据访问逻辑进行编码就能编写应用程序,能生成更轻巧,与数据库无关的应用程序,这种应用程序更易于开发与维护。

3) 分布式共享数据对象 Cache 服务

让开发人员编制高性能的系统,极大提高整体部署的伸缩性。

3. 支持异构环境

J2EE 能够开发部署在异构环境中的可移植程序。基于 J2EE 的应用程序不依赖任何特定操作系统、中间件、硬件。因此设计合理的基于 J2EE 的程序只需开发一次就可部署到各种平台。这在典型的异构企业计算环境中是十分关键的。J2EE 标准也允许客户订购与 J2EE 兼容的第三方的现成组件,把它们部署到异构环境中,节省了由自己制订整个方案所需的费用。

4. 可伸缩性

企业必须选择一种服务器端平台,这种平台应能提供极佳的可伸缩性以满足那些在他们系统上进行商业运作的大批新客户。基于 J2EE 平台的应用程序可被部署到各种操作系统上,例如高端 UNIX 与大型机系统,这种系统单机可支持 64~256 个处理器(这是 NT 服务器所望尘莫及的)。J2EE 领域的供应商提供了更为广泛的负载平衡策略,能消除系统中的瓶颈,允许多台服务器集成部署。这种部署可达数千台处理器,实现可高度伸缩的系统,满足未来商业应用的需要。

5. 稳定的可用性

服务器端平台必须能全天候运转以满足公司客户、合作伙伴的需要,因为 Internet 是全球化的、无处不在的,即使在夜间按计划停机也可能造成严重损失。若意外停机,那会产生灾难性后果。J2EE 部署到可靠的操作环境中,支持长期的可用性。一些 J2EE 部署在 Windows 环境中,客户也可选择健壮性能更好的操作系统,如 Sun Solaris、IBM OS/390。最健壮的操作系统的可用性可达到 99.999% 或每年只需 5min 停机时间。这是实时性很强商业系统理想的选择。

1.3 J2EE 的构成

J2EE 的主要组成部分:

- J2EE 平台(J2EE Platform): J2EE 平台主要就是利用 Java2 平台的标准版来为当前的企业级应用提供解决方案的设计、开发、装配和部署,并定义了一组核心 API 标准的接口。
- J2EE 规范(Platform Specification): 有相应的技术规范并对 API 进行了解释。
- 参考实现(Reference Implementation): 是关于 J2EE 平台语法的正确范例。

- 兼容性测试套件(Compatibility Test Suite): 可以使不同 J2EE 版本的产品运行的程序。
- J2EE 蓝图(J2EE BluePrints): 提供了基于 J2EE 多层应用的实例与文档, 主要参考 PetStore、Adventure Builder。

1.4 J2EE 平台的角色

1. J2EE 产品提供者

J2EE 产品提供者是设计并提供实现 J2EE 规范所定义的 J2EE 平台、API 和其他功能的公司。这一般是指操作系统、数据库系统、应用程序服务器或 Web 服务器的卖主, 它们依照 J2EE 的规范实现 J2EE 平台。

2. 工具提供者

工具提供者是指建立组件提供者、装配者和部署者所使用的开发、装配和打包工具的公司或个人。

3. 应用程序组件提供者

应用程序组件提供者是指建立 J2EE 应用程序所使用的 Web 组件、Enterprise Bean、Applet 或应用程序客户端的公司或个人。

4. Enterprise Bean 开发者

Enterprise Bean 开发者执行下面的任务并递交包含 Enterprise Bean 的 EJB JAR 文件:

- 编写并编译源代码;
- 详细说明部署描述;
- 将 .class 文件和部署描述绑定到一个 EJB JAR 文件中。

5. Web 组件开发者

Web 组件开发者执行下面的任务并提交包含 Web 组的 WAR 文件:

- 编写并编译 servlet 源代码;
- 编写 JSP 和 HTML 文件;
- 详细说明 Web 组件的部署描述;
- 将 .class、.jsp、.html 和部署描述绑定到 WAR 文件中。

6. J2EE 应用程序客户端开发者

应用程序客户端开发者执行下面的任务并提交包含 J2EE 应用程序客户端的 JAR 文件:

- 编写并编译源代码;
- 详细说明客户端的部署描述;
- 将 .class 文件和部署描述绑定到 JAR 文件中。

7. 应用程序装配者

应用程序装配者是从组件提供者接收应用程序组件 JAR 文件, 并将其装配到一个 J2EE 应用程序

EAR 文件中的公司或个人。装配者或部署者可以直接编辑部署说明,或者使用可以根据交互式正确添加 XML 标志的工具。应用程序装配者执行下列任务并递交包含 J2EE 应用程序的 EAR 文件:

- 将上一阶段建立的 EJB JAR 文件和 WAR 文件装配到一个 J2EE 应用程序(EAR)文件中;
- 详细说明有关 J2EE 应用程序的部署;
- 检验 EAR 文件中的组件是否遵守 J2EE 规范。

8. 应用程序部署者和系统管理员

应用程序部署者和系统管理员可以是公司或个人,他们配置和部署 J2EE 应用程序,管理 J2EE 应用程序在其中运行的计算机和网络这些低层结构,并对运行环境进行监控。他们的任务可能包括:设置事务控制、安全属性并指定数据库连接。

在配置时,部署者按照由应用程序组件提供者提供的指示以解决外部的支持,指定安全设置并定义事务属性。在安装时,部署者将应用程序组件装入服务器中并生成特定容器的类和接口。

部署者/系统管理员执行下列任务以安装和配置 J2EE 应用程序:

- 将上一阶段生成的 J2EE 应用程序(EAR)文件添加到 J2EE 服务器中;
- 根据运行环境,通过修改 J2EE 应用程序的部署说明对 J2EE 应用程序进行配置;
- 检验 EAR 文件的内容是否遵守 J2EE 规范;
- 部署(安装)J2EE 应用程序 EAR 文件到 J2EE 服务器中。

1.5 J2EE 体系架构

J2EE 使用多层的分布式应用模型,应用逻辑按功能划分为组件,各个应用组件根据它们所在的层分布在不同的机器上。事实上,Sun 设计 J2EE 的初衷正是为了解决两层模式(Client/Server)的弊端,在传统模式中,客户端实现与服务器的直接相连,没有中间环节,缺少通用性的特点,业务变更或改变不够灵活,需要重新设计和开发,增加了维护和管理的难度,进一步的业务拓展困难较多。现在 J2EE 的多层企业级应用模型将两层化模型中的不同层面切分成许多层。多层化应用能够为不同的每种服务提供一个独立的层,具有分布性特点,可以随时随地进行业务处理。业务扩展简单方便,通过增加网页即可增加服务器功能。维护简单方便,只需要改变网页,即可实现所有用户的同步更新,并且开发简单、共享性强。以下是 J2EE 典型的四层结构(如图 1-1 所示):

- 运行在客户端机器上的客户层组件;

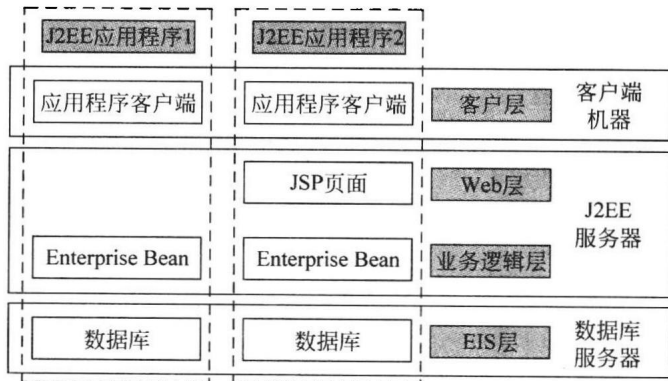


图 1-1 J2EE 四层结构图

- 运行在 J2EE 服务器上的 Web 层组件；
- 运行在 J2EE 服务器上的业务逻辑层组件；
- 运行在数据库服务器上的 EIS(Enterprise Information System, 企业信息系统)层软件。

1. J2EE 应用程序组件

J2EE 应用程序是由组件构成的。J2EE 组件是具有独立功能的软件单元,它们通过相关的类和文件组装成 J2EE 应用程序,并与其他组件交互。J2EE 说明书中定义了以下的 J2EE 组件:

- 应用客户端程序和 Applets 是客户层组件；
- Java Servlet 和 JavaServer Pages(JSP)是 Web 层组件；
- Enterprise JavaBean(EJB)是业务层组件。

2. 客户层组件

比较常用的客户层组件是 Browser 和应用程序客户端。

3. Web 层组件

J2EE Web 层组件可以是 JSP 页面或 Servlet。按照 J2EE 规范,静态的 HTML 页面和 Applets 不算是 Web 层组件。

如图 1-2 所示的客户层那样,Web 层可能包含某些 JavaBeans 对象来处理用户输入,并把输入发送给运行在业务层上的 Enterprise Bean 进行处理。

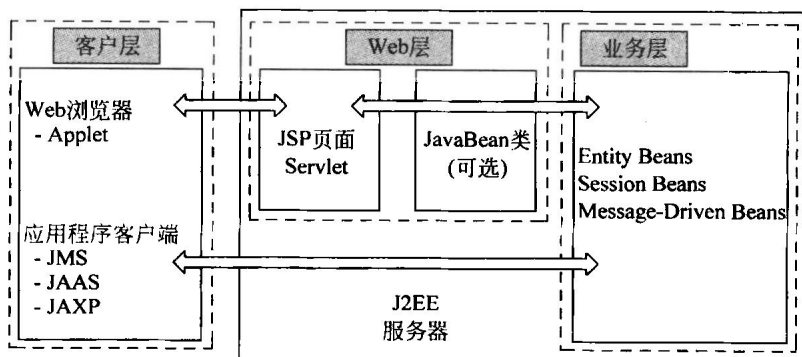


图 1-2 J2EE 三层结构

4. 业务层组件

业务层代码的逻辑用来满足银行、零售和金融等特殊商务领域的需要,由运行在业务层上的 Enterprise Bean 进行处理。图 1-3 表明了一个 Enterprise Bean 是如何从客户端程序接收数据,进行处理(如果必要的话),并发送到 EIS 层储存的,这个过程也可以逆向进行。

有三种企业级的 Beans: 实体(Entity)Beans、会话(Session)Beans 和消息驱动(Message-Driven)Beans。

- 会话 Beans 表示与客户端程序的临时交互。当客户端程序执行完后,会话 Beans 和相关数据就会消失。
- 实体 Beans 与会话 Beans 相反,表示数据库的表中一行永久的记录。当客户端程序中止或服务器关闭时,就会有潜在的服务保证实体 Beans 的数据得以保存。

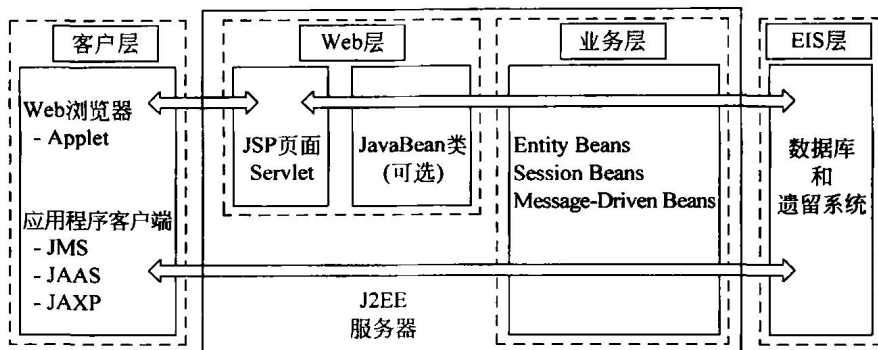


图 1-3 J2EE 四层结构

- 消息驱动 Beans 结合了会话 Beans 和 JMS 的消息监听器的特性,允许一个业务层组件异步接收 JMS 消息。

5. EIS 层

企业信息系统层处理企业信息系统软件包括企业基础建设系统,如企业资源计划(ERP)、大型机事务处理、数据库系统和其他的遗留信息系统。例如,J2EE 应用组件可能为了数据库连接需要访问企业信息系统。

1.6 J2EE 核心技术

J2EE 平台由一整套服务(Services)、应用程序接口(APIs)和协议构成,它对开发基于 Web 的多层应用提供了功能支持,下面对 J2EE 中的 13 种技术规范进行简单的描述。

1. JDBC(Java DataBase Connectivity)

JDBC API 为访问不同的数据库提供了一种统一的途径,像 ODBC 一样,JDBC 对开发者屏蔽了一些细节问题,另外,JDBC 对数据库的访问也具有平台无关性。

2. JNDI(Java Name and Directory Interface)

JNDI 是 Java 命名和目录服务的简称,JNDI 为定位用户、资源与服务提供了一个标准接口。它作为客户端与命名和目录服务的访问接口,隐藏了各种命名和目录服务的细节,使得开发人员可以在各种服务之间进行访问。

JNDI 定义了两组 API,分别为 JNDI API(Application Programming Interface,应用程序接口)和 JNDI SPI(Service Provider Interface,服务供应商接口),Java 应用程序通过 JNDI API 访问各种命名目录服务,服务供应商接口提供了供应商使用的功能。

3. EJB(Enterprise JavaBean)

J2EE 技术赢得媒体广泛重视的原因之一就是 EJB。它们提供了一个框架来开发和实施分布式商务逻辑,由此很显著地简化了具有可伸缩性和高度复杂的企业级应用的开发。EJB 规范定义了 EJB 组件在何时如何与它们的容器进行交互作用。容器负责提供公用的服务,例如目录服务、事务管理、安全