

挑战 Java 程序员系列丛书

免费售后培训

免费电子课件及案例源代码

大量论坛技术支持

Java EJB

应用程序设计

BitService

北京比特塞威斯图书服务事业部 组编

栗松涛 编著



机械工业出版社
CHINA MACHINE PRESS

TP312/2500

:5

2008

挑战 Java 程序员系列丛书

Java EJB 应用程序设计

北京比特塞威斯图书服务事业部 组编

栗松涛 编著

机械工业出版社

本书全面、系统地讲解了 Java EJB 应用程序的设计方法及技巧, 主要包括 Java EE 5 和 EJB 3.0 组件概述, 会话 EJB 的开发方法、组件设计及生命周期, JMS 程序设计与消息队列, 消息驱动 Bean、SOAP 与 Web Service 开发, Java 持久化程序设计, 伪对象数据库、反向依赖注入、切面程序的设计和事务管理等, 最后基于 Oracle 数据库实现了一个高校教室综合信息管理系统, 以此演示使用 EJB 设计应用程序的方法。

本书内容安排由浅入深、由易到难, 通过丰富的教学案例全面讲解了 EJB 的相关技术。本书另一个特点是每章都结合完整程序代码演示特定知识点, 并介绍一些在具体项目开发中的应用技术。本书中的程序不需要特殊运行条件和编程环境, 只需安装 Java EE 5、Java SE 5 或 Java SE 6、JBoss 4.2.1、Apache Ant 1.6.5 和记事本软件即可。读者只要坚持调试通书中的每一段程序, 相信可以全面掌握这门新一代的企业级软件开发技术。

本书适用于 Java EJB 应用程序设计的初学者或具有中等水平的读者, 也可作为各类 Java 培训班的教材。

图书在版编目 (CIP) 数据

Java EJB 应用程序设计 / 栗松涛编著. —北京: 机械工业出版社, 2008.3
(挑战 Java 程序员系列丛书)
ISBN 978-7-111-23754-9

I. J… II. 栗… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 036909 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 吴鸣飞

责任印制: 洪汉军

北京振兴源印务有限公司印刷厂印刷

2008 年 4 月第 1 版·第 1 次印刷

184mm × 260mm · 35 印张 · 866 千字

0001—5000 册

标准书号: ISBN 978-7-111-23754-9

定价: 55.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

销售服务热线电话: (010) 68326294 68993821

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 88379753 88379739

封面无防伪标均为盗版

丛 书 序

在信息技术高速发展的今天,计算机技术已经广泛地应用于各行各业。软件行业作为计算机技术领域的核心产业得到了蓬勃的发展,而软件开发技术作为计算机及其相关行业的关键技术,越来越受到重视。当前,软件开发基本上有两种主流软件开发工具,即 Java 和 .NET。这两种开发工具功能强大、应用领域广泛,是其他开发工具所不能比拟的。可以说,熟练掌握这两种开发工具中的一种,再拥有一定程度的开发经验,就可以成为一名合格的软件开发人员。

目前,相对于大多数职位的人才都面临激烈竞争的情况,合格的软件开发人员却一直是供不应求的。这说明市场上对软件人才需求量很大。但由于软件开发需要丰富的实际工作经验,而对仅仅接受过课本教育,却很少开发实际项目的毕业生来说,是很难立刻胜任软件开发的工作岗位的,从而也就无法达到用人企业的要求,这就是所谓的“人才错位”。而能够有效弥补毕业生与合格的软件开发人员之间差距的途径,目前主要有两种:一是通过上专业培训班或者自己看书学习,并配合相关上机操作自学成才;二是通过用人单位提供实践的机会来获取能力。显然,对刚毕业的学生来说,第一种途径实现起来更容易。

虽然市场上有很多针对 Java 程序设计的参考书和教材,但是,已有的同类书都无法回避的一个事实就是:仅仅依靠纸介质图书和有限的源代码程序来学习 Java 程序设计,不但学习起来非常吃力,而且所得到的结果距离真正的 Java 程序员的需求还有一段距离。我们的结论就是:想真正掌握 Java 这样一种复杂的程序开发工具,除了要学习一套易学易用的图书,还需要看懂大量的实例源代码,以及专家随时的答疑解惑和面对面的学习指导,再加上充足的上机实践。只有这样,才能够真正达到或者接近一个合格的 Java 程序员的水平。

我们开发的这套“挑战 Java 程序员系列丛书”,不但图书本身遵循了由浅入深、循序渐进的学习规律,能够使读者轻松快速地掌握 Java 开发的系统知识,而且还配备了完整的售后服务系统,包括:3 个学时的免费培训服务、免费从网上下载 PPT 电子课件、免费下载实例源代码、论坛、E-mail 技术支持服务等。通过这些全方位的售后服务,将 Java 程序员的必备技能依次讲授给读者,为读者铺就一条通向 Java 程序员的成功之路。

本套丛书十分适合于立志成为一个程序员的大中专院校相关专业的学生,或具有一定基础的软件开发爱好者。本套丛书共 6 个品种,包括《Java 面向对象程序设计》、《Java SE 应用程序设计》、《Java XML 应用程序设计》、《Java 数据库应用程序设计》、《Java Web 应用程序设计》和《Java EJB 应用程序设计》,如图 1 所示。

免费售后服务说明

我们为本套丛书的读者提供了丰富的免费售后服务,包括以下 4 方面。

1. 每本书免费赠送 3 个学时的培训(培训地点在北京)

“挑战 Java 程序员系列丛书”中的每本图书自带了 3 个学时的售后培训,购买该书的读者都可以参加。但需要提前预约,否则培训机构不能提供对应的培训内容。

每本图书的售后服务培训班通常根据图书销售情况,周期性推出新班,并通过网站

(<http://www.bits-service.com>) 发布。读者需要提前到网站查询相应班级, 然后通过网络或电话报名预约, 预约电话为 010-64446364 和 010-64446332。预约完的读者在参加培训前需要到网站上或通过电话确认对应班级的变更情况, 培训机构会取消一些人数不多的班, 并将取消信息显示在网站上, 我们将不再一一通知读者, 请读者朋友认真核实。图 2 显示了报名预约的基本流程。

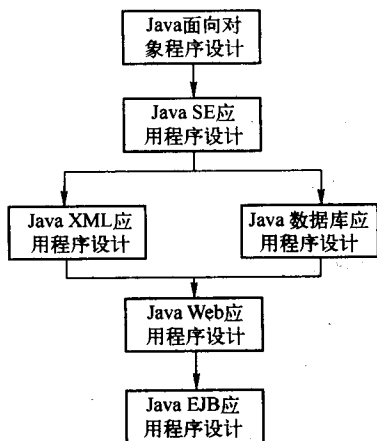


图 1

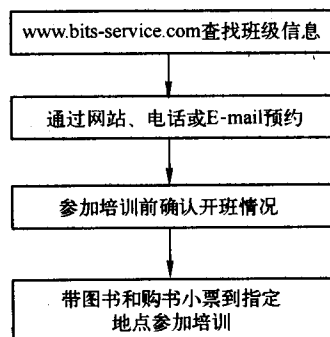


图 2

2. 免费提供精美的 PPT 电子课件

读者可以登录 www.bits-service.com 下载各本图书的电子课件。

3. 免费提供实例源代码

要下载代码的读者, 请首先登录 <http://www.bits-service.com>, 在主页上选择“比特塞威斯图书服务中心 (BITS Book)”, 进入图书服务首页, 在首页菜单条上选择“读者服务”进入“读者服务”频道, 随后可以利用网站的搜索工具或使用标题列表找到对应图书, 在相应的图书网页上单击“源代码下载”超链接, 下载该图书的源代码压缩包。本系列丛书的源代码清单通常按照章节组织, 读者参考图书, 可以使用对应的源代码。

4. 论坛和 E-mail 技术支持

我们为本系列图书提供了专业服务论坛, 网址是 <http://www.bits-service.com> 主页→图书服务论坛→Java 系列图书, 读者在阅读过程中若发现问题, 可以到论坛里发帖, 培训机构的专业老师会定期进行回复。论坛为每本书提供一个讨论区, 建议读者将问题发送到对应图书的讨论区中, 以便作者及时收集和处理。读者也可以发送 Email 给 booksupport@bits-service.com 寻求专家技术支持。

此外, 我们还会提供一部分特级教师的课堂视频教学文件, 读者可以通过网站下载。

增值服务

除了上述提供的免费服务外, 如果读者希望进一步学习, 我们还提供了一部分增值服务, 包括 Java 项目实习服务、Java 程序员就业推荐服务等。这部分服务需要收取一定费用, 读者如果有兴趣, 可以按照以下联系方式与比特塞威斯公司联系。

增值服务咨询邮箱: allsupport@bits-service.com; 增值服务咨询电话: 010-64446368。

前 言

Java EE 是一种成熟的主流企业应用解决方案,而 EJB 技术构成了该解决方案的主体。在 Java EE 1.5 以前的版本中,EJB 技术由于其自身的设计缺陷,导致很多程序员难以熟练掌握,因此被束之高阁。软件行业也随即产生了诸如“没有 EJB 的 Java EE 技术”、“轻量级 Java EE 技术”等舍弃了 EJB 部分的 Java EE 解决方案。Java EE 5 的出现改变了传统 EJB 的设计缺陷,使目前 EJB 组件成为了一种方便的轻量级组件技术,适合于各种企业商务逻辑的处理。EJB 3.0 是一种轻量级组件技术,和其他轻量级组件技术比,EJB 3.0 规范下的组件需要特定服务器支持,组件必须部署到服务器上才能正常运行。这种模式的优点是可以使用服务器提供的各种企业级服务;缺点是组件无法独立使用。本书所有的知识都基于 JBoss 4.2.1 服务器进行开发和调试。

本书针对 Java EE 中的 EJB 开发技术进行了深入讨论,是作者实际开发心得和体会的汇总,很多案例都有很强的项目背景,是读者掌握 EJB 技术的一本不可多得的教材。全书由 8 个部分共 21 章组成。

第 1 部分整体介绍 Java EJB 技术,包括第 1 章。该部分主要讨论 Java EE 技术的特点、EJB 组件的分类、部署环境搭建和开发环境配置。

第 2 部分详细介绍会话 EJB 组件开发技术,包括第 2~4 章。该部分重点对会话 EJB 的特点和开发技术进行了详细讨论。主要包括会话 EJB 的特点、分类、与传统 EJB 组件的区别、开发设计方法、配置部署、异常处理、参数设计和生命周期等内容。该部分的重点主要集中在单个会话 EJB,对多组件相互关联的讨论位于本书第 6 部分。

第 3 部分详细介绍消息驱动 Bean 的相关技术,包括第 5~6 章。其中第 5 章介绍了 JMS 程序设计方法,包括消息队列的分类、部署、配置和程序设计,是学习第 6 章的基础。而第 6 章针对消息驱动 Bean (MDB) 的特点和开发技术进行讨论,包括 MDB 组件的设计方法、生命周期和事件处理等,本章最后结合实际项目,设计了一个项目请求均衡器 (Load Balancer),演示了 MDB 的典型用法。本章主要内容集中于单个 MDB 的讨论,对 MDB 组件和其他组件,如会话组件的相互关联和调用,将在本书第 6 部分进行讨论。

第 4 部分详细介绍 Java EE 开发中的 Web Service 组件技术,包括第 7~10 章。其中第 7 章对 Web Service 技术进行了概述,重点讨论了 Web Service 组件的设计方法,是后面第 8~10 章的基础。第 8 章讨论了 Web Service 标准,包括 WSDL 和 SOAP,并讨论了 Web Service 接口的概念和定义方法,在此基础上简单介绍了 WSDL 接口设计语言。SOAP 是 Web Service 组件的实现协议,本章在介绍 SOAP 标记语法的同时,设计了两种通过 SOAP 消息调用 Web Service 组件的方法。第 9 章介绍 Web Service 组件的设计方法,包括基于类的设计和基于接口的设计,后者在项目开发中具有重要意义。此外本章还讨论了 Web Service 客户端代理和客户程序的基本设计方法。第 10 章讨论了 Web Service 组件的高级特性,包括 Web Service 设计中的接口参数与异常处理原则。

第 5 部分详细介绍 Java 持久化(JPA)程序设计,主要包括第 11~17 章。原则上 JPA 不是 Java EE 规范的组成部分,其 API 同样可以在 Java SE 环境中使用,但为了与传统 EJB 的处

理规则相同,这里通过对这种技术来弥补 EJB 3.0 中没有实体 Bean 的现象。JPA 的本质是为目前通用的 Java ORM 技术提供接口标准,本书讨论的 JBoss 4.2.1 中的 JPA 就是基于 Hibernate 3 的一种实现。JPA 的存在为关系型数据库提供了对象风格的操作接口,这里将其称为伪对象数据库。其中,第 11 章讨论了伪对象数据库的概念,最后讨论了 JPA 在 Java SE 应用环境、会话 EJB 组件环境、Web Service 环境下的使用方法,是后面各章内容的基础。

第 12 章重点介绍单个实体 Bean 的特点、行为、管理方法和生命周期,本章最后在讨论实体 Bean 的基础上,讨论了持久性单元和伪对象数据库的关系。第 13 章重点讨论实体 Bean 的主键映射方法。第 14 章则按照实体 Bean 中属性类型的不同,分别讨论了和数据库中对应字段的关联映射方法,包括枚举型、时间型、LOB 型以及嵌入式自定义对象型等。第 15 章重点讨论实体 Bean 间的关联关系,包括一对多、多对多等关联,在实际项目中具有重要意义。第 16 章综合运用前面内容设计了一个描述学生选课过程的伪对象数据库,演示了使用 JPA 处理实际商务系统的方法。第 17 章在 16 章所设计的伪对象数据库基础上,讨论了 EJB QL 语言,该语言在伪对象数据库中的地位 and 关系型数据库中的 SQL 语句等价,是程序操作伪对象数据库的脚本语言。

第 6 部分详细介绍 EJB 组件集成的技术,主要包括第 18~19 章,其中 18 章详细讲解了 EJB 3.0 的反向依赖注入 (IoC) 技术,这是目前主流的组件设计技术;第 19 章重点讲解了 EJB 3.0 对面向切面程序设计 (AOP) 的支持,主要讨论了切面的概念和基本类型,这两部分构成了将不同 EJB 组件集成为复杂软件系统的机制,在 EJB 平台中具有重要地位。

第 7 部分讨论了 EJB 组件事务设计方法,包括第 20 章。EJB 提供了两种风格的事务调用:声明性事务和程序性事务,前者通过配置文件管理组件行为的事务性,后者通过程序语句控制组件逻辑的事务性,是实际项目设计中必不可少的实用技能。

第 8 部分讨论使用前面章节内容开发 Java EE 项目的方法,包括第 21 章。通过对高校教室综合管理系统的开发,为读者演示基于 Java EJB 开发企业应用的过程和方法,也是本书的内容总结及应用。

本书主要由栗松涛老师负责编写,在编写过程中,比特塞威斯职业教育部和企业培训部的所有老师都为本书大纲的制定和编写提供了很多宝贵意见,部分老师还参与了本书部分章节的整理和案例制作,这里不再一一列出,并由衷地表示感谢。

书中遗漏或错误之处,敬请读者批评指正。读者若对本书有什么疑问,可以发电子邮件到 jsjfw@mail.machineinfo.gov.cn,我们会尽快给予解决。

本书提供了一个专业的服务论坛,读者朋友在阅读过程中发现问题或疑难,可以到该论坛里发帖子,比特塞威斯的专业老师会定期进行回复。论坛的地址是:

<http://www.bits-service.com> 主页→图书服务论坛→Java 系列图书→Java EJB 应用程序设计

北京比特塞威斯图书服务事业部

目 录

丛书序

前言

第 1 章 Java EE 5 和 EJB 3.0 概述 ··· 1

1.1 Java EE 5 概述 ····· 1

1.1.1 Java EE 5 基本架构 ····· 1

1.1.2 Java EE 5 规范 ····· 3

1.1.3 Java EE 5 中的 Web 组件 ····· 3

1.1.4 Java EE 5 中的 EJB 组件 ····· 4

1.2 EJB 3.0 开发环境的安装与配置 ····· 5

1.2.1 Java SE 环境的安装和配置 ····· 5

1.2.2 Java EE 环境的安装和配置 ····· 6

1.2.3 JBoss 环境的安装和配置 ····· 11

1.2.4 Apach Ant 的安装和配置 ····· 14

1.3 本书源代码的安装 ····· 15

1.4 小结 ····· 15

第 2 章 会话 EJB 开发方法概述 ····· 16

2.1 会话 EJB 简介 ····· 16

2.1.1 会话 EJB 的分类 ····· 16

2.1.2 会话 EJB 的结构 ····· 17

2.2 EJB 3.0 无状态会话 EJB 的开发 ····· 18

2.2.1 无状态会话 EJB 的概念 ····· 18

2.2.2 无状态会话 EJB 的开发 ····· 19

2.2.3 无状态会话 EJB 的编译与部署 ····· 22

2.2.4 无状态会话 EJB 的使用和测试 ····· 25

2.3 EJB 2.X 无状态会话 EJB 的开发 ····· 28

2.3.1 无状态会话 EJB 的开发 ····· 28

2.3.2 无状态会话 EJB 的部署 ····· 31

2.3.3 无状态会话 EJB 的使用和测试 ··· 34

2.4 EJB 3.0 有状态会话 EJB 的开发 ····· 35

2.4.1 有状态会话 EJB 的概念 ····· 36

2.4.2 有状态会话 EJB 的开发 ····· 36

2.4.3 有状态会话 EJB 的测试 ····· 38

2.5 EJB 2.X 有状态会话 EJB 的开发 ····· 41

2.5.1 EJB 2.X 有状态会话 EJB 的开发 ····· 41

2.5.2 有状态会话 EJB 编译与部署 ··· 43

2.5.3 有状态会话 EJB 的使用和测试 ····· 45

2.6 EJB 3.0 和 EJB 2.X 的比较 ··· 48

2.6.1 组件结构的优化 ····· 48

2.6.2 API 的污染性 ····· 49

2.7 小结 ····· 51

第 3 章 会话 EJB 组件设计 ····· 53

3.1 多接口会话 EJB 组件 ····· 53

3.1.1 多接口会话 EJB 组件开发 ····· 53

3.1.2 多接口会话 EJB 组件的使用 ··· 56

3.2 无污染 EJB 组件设计 ····· 58

3.2.1 非正规接口实现 ····· 58

3.2.2 使用类注解 (Attribute) 法设计 EJB 组件 ····· 59

3.2.3 使用部署文件开发 EJB 组件 ··· 63

3.2.4 EJB 组件开发方法比较 ····· 67

3.3 EJB 接口参数设计 ····· 68

3.4 EJB 接口异常设计 ····· 71

3.4.1 EJB 组件接口中的应用异常 ··· 72

3.4.2 EJB 组件接口中的系统异常 ··· 77

3.5 EJB 组件的配置文件 ····· 82

3.5.1 EJB 组件配置文件格式和读取语法 ····· 82

3.5.2 EJB 组件配置文件读取实例 ··· 84

3.6 小结 ····· 88

第4章 会话 EJB 组件生命周期	89	6.1.2 MDB 组件属性	137
4.1 无状态会话组件生命周期	89	6.2 MDB 组件开发方法	140
4.1.1 无状态会话 EJB 的 3 种状态	89	6.2.1 基于标注的 MDB 设计	141
4.1.2 无状态 EJB 组件生命事件	90	6.2.2 基于配置文件的 MDB 设计	146
4.1.3 无状态 EJB 组件生命事件 处理	91	6.2.3 两种开发方法的比较	150
4.1.4 无状态 EJB 组件生命事件 实例	92	6.3 Topic 型 MDB 组件设计	150
4.2 有状态会话组件生命周期	97	6.4 MDB 组件生命周期	154
4.2.1 有状态会话 EJB 的生命周期	97	6.4.1 消息驱动 EJB 的 3 种状态	154
4.2.2 有状态 EJB 生命周期中的 事件	99	6.4.2 MDB 组件生命事件	155
4.2.3 有状态 EJB 生命事件处理	100	6.4.3 MDB 生命事件处理	156
4.2.4 有状态 EJB 生命事件处理 实例	101	6.5 MDB 负载均衡器设计	162
4.3 小结	106	6.6 小结	167
第5章 JMS 与消息队列	107	第7章 Web Service 概述	168
5.1 JMS 概述	107	7.1 Web Service 的基本特征	168
5.1.1 JMS 消息类型	107	7.1.1 Web Service 组件模型	168
5.1.2 JMS 基本模型	108	7.1.2 Web Service 组件访问模型	169
5.1.3 JMS 消息队列操作模式	110	7.2 设计第一个 Web 服务	170
5.2 JBoss MQ 简介	111	7.2.1 Web Service 组件设计	170
5.2.1 在 JBoss MQ 上创建队列	111	7.2.2 Web Service 组件的编译与 部署	171
5.2.2 在 JBoss MQ 上创建主题	113	7.2.3 Web Service 客户代理	173
5.3 JMS 消息队列操作模型	114	7.2.4 Web Service 客户程序	175
5.3.1 JMS 消息操作框架	114	7.3 Java Web Service 技术小结	177
5.3.2 JMS 消息操作对象体系	115	7.4 小结	178
5.3.3 JMS 消息操作对象体系的 创建	116	第8章 WSDL 接口设计与 SOAP 协议	179
5.4 JMS 1.0.2 消息队列操作	117	8.1 本章技术准备	179
5.4.1 PTP 消息队列操作接口	117	8.2 WSDL 文件整体结构	180
5.4.2 主题消息的发布与订阅	126	8.3 接口数据类型	181
5.5 JMS 1.1 消息队列操作	130	8.4 WSDL 接口	184
5.5.1 消息的发送	130	8.4.1 接口要素	184
5.5.2 消息接收	131	8.4.2 接口实现细节	185
5.6 小结	134	8.5 WSDL 文件生成方法	188
第6章 消息驱动 Bean(MDB)	135	8.5.1 通过 SDK 工具创建 WSDL 文件	188
6.1 MDB 组件概述	135	8.5.2 通过发布生成 WSDL	190
6.1.1 MDB 基本结构	135	8.6 SOAP 基本语法	191
		8.7 通过 SOAP 使用 Web Service	192

8.7.1 通过 Java 语言调用 Web Service	192	11.3.3 操作伪面向对象数据库	245
8.7.2 通过 JavaScript 平台调用 Web Service	195	11.3.4 程序的编译运行	249
8.8 小结	197	11.4 在会话 EJB 中使用 JPA	250
第 9 章 Web Service 设计初步	198	11.4.1 Java EE 中伪对象数据库 构建	250
9.1 Web Service 设计	198	11.4.2 会话 EJB 对 JPA 的调用	253
9.1.1 使用 WebMethod 标注定义 Web 服务接口	198	11.4.3 测试程序设计	255
9.1.2 对 Web Service 接口进行 定制	200	11.5 在 Web Service 中使用 JPA	257
9.1.3 基于接口的 Web Service	204	11.5.1 Web Service 对 JPA 的调用	258
9.2 Web Service 调用方法	206	11.5.2 测试程序设计	259
9.3 可移植 Web Service 调用	209	11.6 小结	260
9.4 小结	211	第 12 章 JPA 实体 Bean	261
第 10 章 Web Service 组件高级 特性	212	12.1 实体 Bean 基本映射	261
10.1 Web Service 参数传递 原则	212	12.1.1 实体 Bean 对应的表结构	261
10.2 Web Service 参数定义	213	12.1.2 基本映射	262
10.2.1 自定义参数	213	12.1.3 表和列的映射	266
10.2.2 自定义参数类型传递	218	12.2 JPA 实体 Bean 的操作	272
10.3 Web Service 集合类型	220	12.2.1 实体对象检索	273
10.3.1 基于数组的参数传递	220	12.2.2 实体对象的状态更新	274
10.3.2 基于链表的参数传递	225	12.2.3 实体对象的保存	276
10.4 Web Service 异常	227	12.2.4 实体对象的删除	278
10.5 小结	232	12.2.5 对象的合并与刷新	279
第 11 章 Java 持久化与对象 数据库	233	12.3 JPA 实体 Bean 生命周期	281
11.1 对象数据库与 JPA	233	12.3.1 实体 Bean 状态特点	281
11.1.1 面向对象数据库(ODDB)	234	12.3.2 持久对象生命事件	284
11.1.2 伪对象数据库与 ORM 软件	234	12.4 持久化单元与实体 Bean	289
11.1.3 ORM 软件与 JPA	235	12.4.1 持久化单元的基本概念	289
11.2 JPA 简介	236	12.4.2 位置关联持久化单元中的 Bean	290
11.3 JPA 程序初步	239	12.4.3 实体 Bean 独立部署	291
11.3.1 伪对象数据库与持久化 单元	239	12.5 小结	298
11.3.2 设计伪对象数据库结构	240	第 13 章 实体 Bean 主键映射	299
		13.1 标识主键	299
		13.1.1 手工设置型主键映射	299
		13.1.2 Identity 主键映射	300
		13.1.3 Sequence 主键映射	304
		13.1.4 TableGenerator 主键映射	305
		13.2 复合主键映射	310

13.2.1 数据库表结构定义	310	第 16 章 基于 JPA 的伪对象数据库	
13.2.2 使用 IdClass 映射复合主键 ...	311	设计	400
13.2.3 EmbeddedId 主键映射	316	16.1 伪对象数据库与域模型	400
13.3 小结	321	16.1.1 域模型中的实体设计	401
第 14 章 实体 Bean 属性映射	322	16.1.2 伪对象数据库的设计	402
14.1 实体 Bean 数据加载规则 ...	322	16.2 伪对象数据库底层结构	409
14.2 基本属性映射	328	16.3 伪对象数据库测试	411
14.3 LOB 属性映射	332	16.4 小结	414
14.3.1 批量映射模式	333	第 17 章 对象查询语言 EJB QL	415
14.3.2 缓存映射	338	17.1 EJB QL 检索语句	416
14.3.3 对象序列化	344	17.1.1 对象检索语句	416
14.3.4 基于 XML 的 LOB 字段		17.1.2 对象属性检索语句	416
映射	348	17.1.3 distinct 与 orderby 子句	417
14.4 自定义对象映射	349	17.2 EJB QL 条件语句	418
14.4.1 自定义枚举映射	349	17.2.1 between、in 与 like 语句	418
14.4.2 自定义非持久类映射	354	17.2.2 is null、is empty 和 member of	
14.5 小结	359	语句	419
第 15 章 实体 Bean 关联映射	360	17.2.3 子查询	420
15.1 单一 Bean 实体属性映射 ...	360	17.3 对象操作语句	420
15.1.1 实体对象模型	360	17.4 EJB QL 执行 API	421
15.1.2 对象在关系型数据库中的		17.4.1 JPA 数据库操作 API	421
存储结构	364	17.4.2 对象分页检索	423
15.1.3 关联关系测试	365	17.4.3 对象属性检索	426
15.1.4 XML 关联映射	367	17.4.4 对象操作	427
15.2 JPA 级联特性	368	17.5 EJB QL 参数	430
15.3 集合属性映射	369	17.6 EJB QL 本地查询	432
15.3.1 普通集合映射	370	17.7 使用配置文件管理查询	
15.3.2 链表集合映射	376	脚本	435
15.3.3 Map 集合映射	379	17.8 小结	438
15.4 双向一对多关联映射	383	第 18 章 EJB 反向依赖注入	439
15.4.1 双向多对一实体关联	383	18.1 反向依赖注入	439
15.4.2 双向关联的对称性问题	386	18.1.1 依赖与依赖注入	439
15.4.3 对双向多对一关联的测试 ...	387	18.1.2 反向依赖注入原则	441
15.5 双向多对多映射	390	18.1.3 反向依赖注入设计方法	
15.5.1 对象在数据库中的存储		概述	443
结构	391	18.2 EJB 反向依赖注入案例	446
15.5.2 实体间的双向多对多关联 ...	392	18.3 JNDI 全局注册表	450
15.5.3 双向多对多关联实体操作 ...	396	18.4 EJB 反向依赖注入机制	452
15.6 小结	398	18.4.1 ENC 的基本概念	452

18.4.2	ENC 注册记录管理	453	20.3	程序性事务	499
18.4.3	ENC 注册记录的检索	456	20.4	EJB 组件的事务特性	502
18.5	EJB 注入点类型	459	20.4.1	无状态会话组件事务特征 ...	502
18.5.1	持久化单元注入	459	20.4.2	MDB 组件事务	502
18.5.2	资源型注入	464	20.4.3	有状态会话 EJB 组件事务 ...	503
18.6	小结	469	20.4.4	实体 Bean 组件事务	508
第 19 章	EJB 切面程序设计	470	20.5	小结	511
19.1	切面的基本概念	470	第 21 章	高校教室综合信息管理	
19.2	EJB 3.0 切面类型	476		系统	512
19.2.1	生命周期事件切面	476	21.1	系统基本需求	512
19.2.2	切面的变体	479	21.2	系统概要设计	514
19.3	EJB 3.0 切入点定义方法	482	21.3	系统开发模块划分	517
19.4	切面行为特征	484	21.4	商务逻辑层设计	521
19.4.1	切面组件反向依赖注入	484	21.5	客户端软件设计	525
19.4.2	切面逻辑中的异常处理	489	21.5.1	Web Service 代理模块	526
19.5	小结	491	21.5.2	Classroom 实体管理界面	
第 20 章	事务	492		模块	528
20.1	事务的基本特征	492	21.5.3	客户端系统的版权界面	
20.2	声明性事务	494		模块	540
20.2.1	声明性事务案例	494	21.5.4	客户端系统引导模块	542
20.2.2	声明性事务模式	497	21.6	小结	542
20.2.3	声明性事务的控制	498	附录	软件调试环境	543

第1章 Java EE 5和EJB 3.0概述

Java EE 5 相对于以前的 Java EE 版本进行了根本性的改变，提供了更为友好的开发方法。这种方法可以和目前流行的 Struts+Spring+Hibernate 的开发模式媲美，是 Sun 公司在结合传统 Java EE 和 Struts+Spring+Hibernate 优点的基础上，重新设计的一套 Java 企业级的解决方案。Java EE 5 剔除了很多传统 EJB 组件中的晦涩部分，是目前十分受欢迎的开发方法。本章重点讨论 Java EE 5 平台的最新特性，并在此基础上讨论 EJB 3.0 组件特点，最后讨论 Java EE 5 最小平台的配置方法，该平台是本书所有案例开发演示的平台。

1.1 Java EE 5 概述

Java EE 5 是基于层次式架构设计的一种企业级系统解决方案，具备 Java 技术的通用特点，包括标准性、开源性和普及性。Sun 公司通过搜集、发掘各企业的开发与集成需求，提出设计规范，并提交各主要软件开发厂商，如 Oracle、IBM、BEA 等公司，在多方参与、协商一致的前提下，根据设计规范，制定某个技术的设计标准，并由 Sun 公司给出标准的参考实现、兼用性测试条件等，由其他厂商开发推广具体 Java 产品。Java EE 5 也是这种流程的产物。

Java EE 5 只是针对企业级应用推出的开发解决方案，但它并不能解决软件开发中的所有问题。例如，如果开发一个高校学生信息管理系统，Java EE 5 则是一个十分合适的解决方案；而如果要开发一个操作系统，Java EE 5 则没有任何的长处，这是由 Java EE 解决方案的特点决定的。Java EE 5 是一种面向 Web 的分布式多层架构平台，适合开发电子商务和信息管理类系统。下面首先讨论 Java EE 5 的基本架构和规范；然后讨论 Java EE 5 中的 Web 组件、EJB 组件。

1.1.1 Java EE 5 基本架构

Java EE 5 是一种典型的多层架构解决方案，这里，层的概念可以通俗理解为代码边界，系统被分成的不同的层，就意味着系统中的代码被分成了很多模块，每个代码属于其中一层。Java EE 5 解决方案典型的层次结构如图 1-1 所示，分为数据存储层、商务逻辑层、展现层，下面分别就各层进行讨论。

1. 展现层

展现层是整个软件系统的外在表现形式，是整个软件中被用户使用的部分。通俗地讲，一个软件系统在运行时，凡是能够被用户看到的软件部分都是系统展现层，都应该被设计到展现层中，系统的展现层重点用来处理软件系统和用户间的交互过程。常用展现层设计技术有 Awt 和 Swing 技术，通过该技术可以设计常见的窗口型应用程序；也可以使用 JSP、HTML、Servlet 等技术实现系统界面，这种系统具备网页类型的界面风格，前者称为 C/S 型应用程序；后者称为 B/S 型应用程序。有关 Awt 和 Swing 等桌面程序设计技术请参考本系列丛书中《Java SE 应用程序设计》一书中的相关论述。有关 HTML、JSP、Servlet 技术的相关内容，在本系列丛书中的《Java Web 应用程序设计》一书中的进行了详细论述。

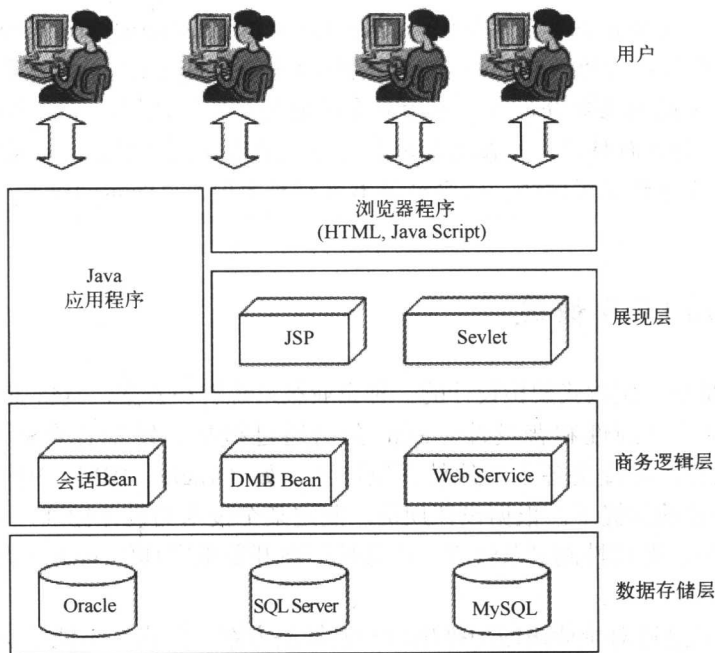


图 1-1 Java EE 5 架构方案

2. 商务逻辑层

商务逻辑层是商务软件系统的核心，表示整个软件所承载的商务规则和商务算法。在 Java EE 规范中，该部分功能通常通过基于无状态会话 Bean 的 Web Service 模式、会话 EJB、消息驱动 EJB 等组件来封装，对应了系统中的商务层逻辑，该部分内容是本书重点，将在后面各章进行详细讨论。

3. 数据存储层

数据存储层主要包括关系型数据库系统，该系统通常不属于 Java EE 范畴，而属于企业应用系统范畴。Java EE 解决方案提供了丰富的集成数据库系统的技术和方法，例如 JDBC、ORM（Object Relation Mapping）、JPA（Java Persistence API）等，都是十分成熟的操作技术，其中 JDBC 和 ORM 在本系列丛书中的《Java 数据库应用程序设计》中进行了详细介

绍, 而 JPA 技术将在本书中进行详细讲解。系统的数据存储层用于存放系统中所要处理的多数信息, 这些信息对应了软件系统的核心, 是企业级信息系统开发中的重要技术。本书和数据库相关的程序都在 Oracle 10g 数据库上进行了调试, 这里需要读者提前掌握 Oracle 10g 数据库的相关知识, 读者可以提前阅读本系列丛书中的《Java 数据库应用程序设计》, 该书提供了本书所需要的操作 Oracle 10g 数据库的相关知识。

1.1.2 Java EE 5 规范

Java EE 本质上只定义了使用 Java 开发软件产品的规则, 其本身并不是软件产品。这些规则称为 Java EE 规范或 Java EE 开发标准。这种特点也注定了基于 Java EE 规范所开发的软件产品在各种 Java 平台间的通用性, 例如基于 JBoss 开发的软件产品可以被成功部署到 WebLogic 平台上, 反之亦然。针对图 1-1 中的 Java EE 多层架构, Sun 公司联合世界上主要软件厂商推出了规范各层技术的 Java 标准, 相关技术标准见表 1-1。

表 1-1 Java EE 标准列表

标准号	标准名称	解 释
JSR-220	Enterprise JavaBeans(EJB) 3.0	EJB 组件的相关规范
JSR-252	Java Server Faces(JSF) 1.2	Web 界面实现框架
JSR-154	Servlet 2.5	Servlet 组件标准
JSR-245	Java Server Page(JSP) 1.2	JSP 标准
JSR-224	Java API for Web Services(JAX-WS) 2.0	Java Web Service 相关标准
JSR-181	Web Service Metadata	Java Web Service 相关标准
JSR-222	Java Architecture for XML Binding 2.0	基于 XML 的分布式调用标准
JSR-220	Java Persistence 1.0	对象持久化标准
JSR-250	Common Annotations 1.0	标注标准
JSR-173	Streaming API for XML (StAX) 1.0	XML 序列化标准

Sun 公司在推出技术标准时, 通常会给出该标准的一个技术参考实现软件, 通过该软件, Java 技术厂商可以提供自己的技术产品, 实现对 Java 标准的支持和推广。读者只要对通用 Java 技术进行学习即可, Java 标准的细节可以通过相关 Java 技术的学习而获得。例如 JSR-154 所对应的 Servlet 技术标准 Servlet 2.5, 其主要内容对应本系列丛书中的《Java Web 应用程序设计》中有关 Servlet 组件开发技术部分。

Sun 公司除推出 Java 技术规范和对对应规范的参考实现外, 通常还要为特定规范提供测试套件 (Test Suite)。通过这些测试套件可对特定 Java 软件进行测试, 成功通过测试的 Java 产品就能够兼容 Java EE 规范下开发的所有软件, 该 Java 产品也称为 Java EE 兼容产品。

1.1.3 Java EE 5 中的 Web 组件

Java EE 5 规范中很重要的一个组成部分是 Web 组件, 也就是本系列丛书中的《Java Web 应用程序设计》所讨论的内容, 在标准上对应了 Java Servlet、Java Server Page (JSP) 和 JSF 等。

Servlet 是一种基于请求响应模式为客户端提供服务的 Web 组件, 其能够接收来自浏览

器的请求,并将处理结果发送给 Web 浏览器。Servlet 需要在支持 Web 的 Java 服务器上运行,例如 Tomcat,并不需要 Java 的企业级服务器,如 WebLogic 等。Servlet 只是一种普通 Web 请求组件,适合于处理普通 Web 请求与调度,而并不适合于实现复杂的商务逻辑。Servlet 组件的一个典型应用是 MVC 设计模式中的控制器。

JSP 是一种简单的设计 Servlet 组件的模式,本质上也是 Servlet 组件。JSP 是通过 HTML 与 Java 代码混合实现 Servlet 组件界面设计与逻辑事件设计的折中方案。JSP 十分适合于软件系统界面的实现,甚至多数公司的 JSP 界面都由美术设计人员而非程序员来完成,由此可见 JSP 在软件系统界面设计方面的优势。

JSF 是一个 Java Web 框架,通俗地讲就是使用 Java Web 技术开发应用程序的“半成品”。JSF 使用 JSP 和 Servlet 对软件系统界面展现中的 MVC 模式提供了典型实现,借助该框架可以开发很多 Java Web UI 组件,使 Java Web 开发类似于 Java 桌面应用程序开发,从而提高 Java Web 系统开发的效率。从 Java EE 1.5 规范之后,JSF 成为 Java EE 技术中的重要组成部分。

Java Web 技术在整个 Java EE 架构中用来实现浏览器风格的应用软件展现,是系统 UI 展现层的典型实现技术,本书重点讨论 EJB 组件的实现,有关 Web 开发技术请参考本系列丛书中《Java Web 应用程序设计》一书的相关论述。

1.1.4 Java EE 5 中的 EJB 组件

EJB 组件是 Java EE 5 平台的重要组成部分。EJB 3.0 规范刻画了 EJB 组件的实现方法,同时也规定了 EJB 运行平台所应该实现的功能和服务,例如 JBoss 服务器的开发就必须满足 EJB 3.0 规范。EJB 3.0 标准实现了 EJB 组件开发者和 EJB 服务器开发者之间的良好契约和分工,基于这种标准,组件开发者所开发的 Bean 可以成功在服务器开发者所开发的平台上运行。

1. EJB 3.0 中的组件类型

EJB 3.0 组件重点包括会话 Bean、消息驱动 Bean、Web Service 组件和 JPA 实体 Bean 等组件。会话 Bean 是商务逻辑的最佳载体,使用会话 Bean 可以实现几乎所有背景下的业务逻辑。消息驱动 Bean 和 Web Service 组件为商务逻辑提供了两种基于消息的松散耦合开发技术,前者基于常用的 TCP/IP 协议,后者通常基于 HTTP/SOAP 协议。该部分内容是本书的重点,本书后面章节会详细地讨论各个组件的特点和开发技术。

2. EJB 服务器所需要提供的服务

使用 EJB 组件开发商务逻辑的最大优点是可以将与业务无关的系统需求,例如事务、消息、持久化、名称查询等服务委托给 EJB 组件的运行平台,使程序员组件开发人员可以将精力集中于商务逻辑的开发,这是使用 Servlet 等技术开发商务逻辑所不具备的优势。

事务是企业级软件开发中最为基本的一项程序功能,通过程序设计的方式可以保证商务逻辑的事务性,例如《Java 数据库应用程序设计》中的 JDBC,但通过程序设计方法实现事务需求,通常比较繁琐,而且开发和维护的工作量都很大。Java EE 5 规范下的 EJB 服务器通常都提供了声明式的事务服务,可以在不修改 EJB 组件代码的前提下,配置管理组件逻辑的事务特征,极大地提高组件的开发和维护效率。在 EJB 程序设计中也可以通过 JTA (Java Transaction API) 接口,以程序方式管理组件逻辑中的事务,尽管如此,其设计方法也比

JDBC 中直接的事务技术简单很多。

JMS 消息服务 (Java Message Service) 也是 EJB 服务器必须提供的基本服务, 通常包括队列式消息服务和广播式消息服务。使用消息服务可以设计面向消息的松散耦合系统, MDB (Message Driven Bean) 就是 EJB 通过消息队列提供服务的组件形式。

Web Service 也是 Java EE 服务器提供的一种有用组件形式, 是目前十分流行的面向服务架构 SOA (Service Oriented Architecture) 的基本技术载体。Java EE 中用于 Web Service 的规范为 JAX-WS (Java API for Web Services), 该规范定义了 Java EE 中 Web Service 的组件类型: 一种通过无状态会话 Bean 的形式提供; 另一种通过 Servlet 形式提供, 前者适合于将一些商务逻辑接口发布成为 Web 服务, 后者则提供了通过 Servlet 定义 Web 服务的最基本形式。

Java 名称解析服务 JNDI (Java Naming and Directory Interface) 也是所有 Java EE 服务器必须提供的一种企业级服务。EJB 服务器通常要将所有组件资源注册到 JNDI 服务器中, 程序开发人员通过 JNDI 服务器的查询可以确定唯一的一个 EJB 组件。JNDI 服务为 EJB 服务器中的全部资源提供了全局唯一的命名管理机制, 是 EJB 服务器的重要组成部分。

EJB 企业服务不是本书的讨论重点, 在用到时本书将进行简单补充, 以满足本书相关知识的论述需要, 感兴趣的读者请参考其他技术资料。

▶▶ 1.2 EJB 3.0 开发环境的安装与配置

EJB 组件的正常编译和部署需要 4 个主要的软件环境: Java SE 1.5 平台、Java EE 1.5 平台、JBoss 4.2.1 服务器和 Apache Ant 1.6.5。特别是 JBoss 服务器的版本号不能太旧, 否则可能不支持目前的 EJB 3.0 规范, Java EE 类库也必须是在 1.5 版之后, 否则没有办法支持本书所讨论的 EJB 3.0 组件, 这些软件都是免费的软件, 可以从相关厂商网站上自由下载, 本书的附录中有具体下载方法的论述。

▶▶▶ 1.2.1 Java SE 环境的安装和配置

Java SE 是所有 Java 软件开发的必须平台, EJB 的开发也不例外, 本书所有的例子都是在 Java SE 1.5/1.6 环境下调试通过的。原则上 EJB 3.0 程序的开发使用 Java SE 1.5 以后的版本即可, 但建议读者安装 Java SE 1.6.0 更新的版本。Java SE 可以从 Sun 的站点上自由下载。安装完成后对应的文件系统结构如图 1-2 所表示。

图 1-2 是 Java SE 1.6 安装完成后的文件夹结构图, 其中 bin 文件夹中是所有的 Java SE 工具, 例如编译工具、打包工具和运行工具等; lib 文件夹是 Java SE 1.6 所包含的基本类库。

安装完成后在系统环境变量中配置 CLASSPATH 和 PATH 环境变量, 将 Java SE 类包中的 dt.jar 和 tools.jar 添加到 CLASSPATH 环境变量中, 并将当前 Java SE 软件包的 bin 路径添加到系统的 PATH 环境变量中。具体安装和配置方法请参考本系列丛书《Java 面向对象程序设计》第 1 章的详细论述, 这里不再赘述。