

# Java EE企业级架构 开发技术与案例教程

杨树林 胡洁萍 编著



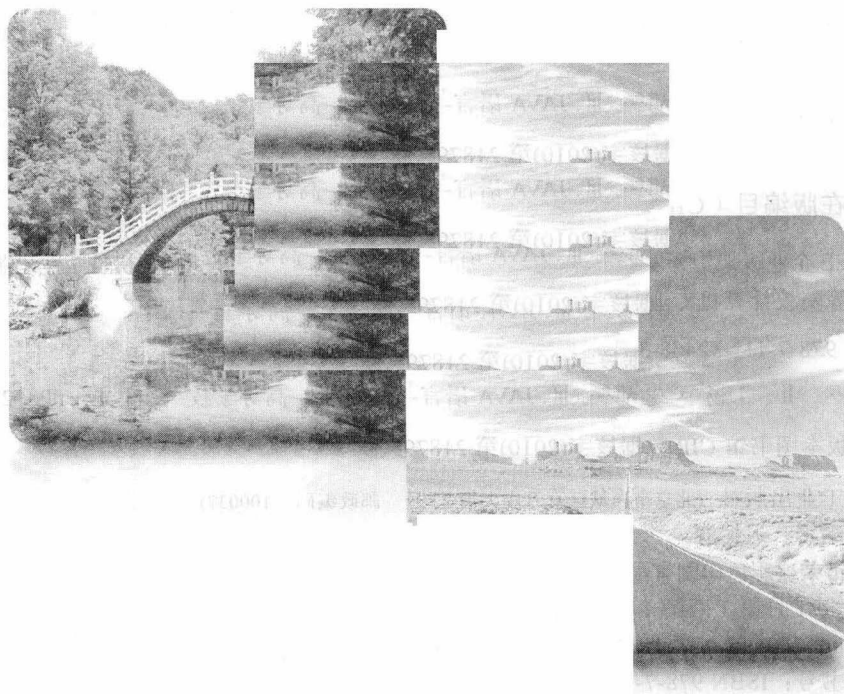
机械工业出版社  
China Machine Press

高等院校计算机专业人才培养规划教材

(应用型)

# Java EE企业级架构 开发技术与案例教程

杨树林 胡洁萍 编著



机械工业出版社  
China Machine Press

Java EE 是目前最为流行的主流技术之一，本书将带领读者进入 Java EE 技术世界，由浅入深地学习各项知识。全书共 8 章，主要内容包括：Java EE 概述、Java Web 应用开发基础、MVC 框架技术 Struts2、持久化技术 JPA、轻量级框架 Spring、EJB 架构、JavaServer Faces 技术、基于 EJB 架构的论坛系统设计等。

本书适合作为大中专院校计算机及其相关专业的教材或参考书，也适合软件开发人员及其他有关人员作为技术参考书。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

### 图书在版编目 (CIP) 数据

Java EE 企业级架构开发技术与案例教程/杨树林, 胡洁萍编著. —北京: 机械工业出版社, 2011.1  
(高等院校计算机专业人才培养规划教材 (应用型))

ISBN 978-7-111-32468-3

I. J… II. ①杨… ②胡… III. JAVA 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2010)第 218795 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: 迟振春

北京市荣盛彩色印刷有限公司印刷

2011 年 1 月第 1 版第 1 次印刷

185mm×260mm • 18.75 印张

标准书号: ISBN 978-7-111-32468-3

定价: 35.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991; 88361066

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzjsj@hzbook.com

## 出版者的话



机械工业出版社华章公司多年来以“全球采集内容，服务中国教育”为己任，致力于引进国际知名大学广泛采用的计算机、电子工程和数学方面的经典教材，出版了一大批在计算机科学界享誉盛名的专家名著与名校教材，其中包括 Donald E.Knuth、Alfred V. Aho、Jim Gray、Jeffery D. Ullman 等名家的一批经典作品。这些作品为我国计算机教育及科研事业的发展起到了积极的推动作用。

近年来，我们一直关注国内计算机专业教育的发展和改革并大力支持、参与相关的教学研究活动。2006年，教育部高等学校计算机科学与技术专业教学指导分委员会在对我国计算机专业教育现状和社会对人才的需求进行研究的基础上，发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》（以下简称《规范》）。为配合《规范》的实施和推广，我们出版了“面向计算机科学与技术专业规范系列教材”。这套教材的推出，对宣传《规范》提出的“按培养规格分类”的理念、推进高校学科建设起到了一定的促进作用。

2007年，教育部下发了《关于进一步深化本科教学改革全面提高教学质量的若干意见》，强调高等教育以育人为本，以学生为主体，坚持以培养创新人才为重点，下大力气深化教育教学改革。在“质量工程”的思想指导下，各高校纷纷开展了相关的学科改革和教学研究活动。高等学校计算机科学与技术专业的教育开始从过去单纯注重知识的传授向注重学科能力的培养转型。2008年年底，教育部高等学校计算机科学与技术专业教学指导分委员会成立了“高等学校计算机科学与技术专业人才培养”项目研究小组，研究小组由蒋宗礼教授（组长）、王志英教授、岳丽华教授、陈明教授和张钢教授组成，研究计算机专业人才培养的构成和在计算机专业的主干课程中如何培养这些专业能力。

为配合“高等学校计算机科学与技术专业人才培养”专项研究成果的推广，满足高校从知识传授向能力培养转型的需求，在教育部高等学校计算机科学与技术专业教学指导分委员会专家及国内众多知名高校专家的指导下，我们策划了这套“高等院校计算机专业人才培养规划教材”。这套教材以专项研究的成果为核心，围绕计算机专业本科生应具有的能力组织教材体系。本套教材的作者长期从事教学和科研工作，他们将自己在本科生能力培养方面的经验和心得融入教材的编写中，力图通过理论教学及实践训练，达到提升本科生专业能力的目标。希望这些有益的尝试能对推动国内计算机专业学生的能力培养起到积极的促进作用。

## 丛书序言

我国高等学校计算机专业建立于20世纪50年代。经过近60年的迅速发展,经历了从精英化教育到大众化教育的发展阶段,目前在在校生多达40余万人,已成为我国规模最大的理工科专业,为国家建设培养了大批信息技术人才。2006年,教育部计算机科学与技术专业教学指导委员会发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范(试行)》(以下简称《规范》),提出了以“按培养规格分类”为核心思想的专业发展建议,把计算机专业人才划分为研究型、工程型、应用型三种不同类型。在《规范》的方针指导下,培养合格的计算机本科人才。

教育包括知识、能力、素质三个方面。知识是基础、载体和表现形式,能力是技能化的知识及其综合体现,素质是知识和能力的升华。专业教育不仅要重视知识的传授,更应突出专业能力的培养,实施能力导向的教育。如何以知识为载体实现能力的培养和素质的提高,特别是实现专业能力和素质的提高是非常重要的。对计算机专业本科教育而言,要想实现能力导向的教育,首先要分析专业能力的构成并考虑如何将其培养落实到教学实践中。为此,教育部高等学校计算机科学与技术专业教学指导委员会开展了计算机科学与技术专业人才培养能力(简称为计算机专业能力)的培养研究。该项研究明确计算机专业本科人才应具有的四项基本能力——计算思维能力、算法设计与分析能力、程序设计与实现能力、系统能力,并将这四大基本能力分解为82个能力点,探讨如何面对不同类型学生的教育需求,在教学活动中进行落实。

针对计算机应用型人才的培养,由于其培养数量巨大、社会需求广泛和多样化,所以培养应用型专业的专业能力在具体教学实践上有其自身的特点。计算机应用型人才培养目标是为国家、企事业信息系统的建设与运行培养信息化技术型人才。本类型人才应能承担信息化建设的核心任务,掌握各种计算机软、硬件系统的性能,善于进行系统的集成和配置,有能力管理和维护复杂信息系统的运行,研究如何实现服务及方便有效地利用系统进行计算等。计算机应用型人才的培养凸显了职业特征,使企业与学校的合作更加紧密,部分课程设置凸显能力培养特征,教学模式也呈现了职业化趋势。

为体现研究成果在教学活动中的实现,我们根据《高等学校计算机科学与技术专业人才培养能力构成与培养》和计算机应用型人才培养的特点和社会需求出版了这套教材。本套教材面向高等院校计算机应用型人才培养从知识传授向能力培养转型的需求,在内容的选择、体系安排和教学方法按照专业能力和职业特征的需要进行了探索和诠释。

本套教材在体系结构上,遵从公共基础课程平台、专业核心课程平台、专业选修课程平台、方向课程平台和基本素质课程平台的体系。专业核心课程主要有程序设计基础、离散数学、数据结构、计算机组成原理、操作系统原理、计算机网络原理、数据库系统原理、编译原理等课程。方向课程分为计算机网络、软件工程、信息系统、程序设计、电子商务、嵌入

式系统、多媒体技术和计算机硬件等方向。在教材编写上，汇集作者才智，重点突出对计算机应用能力和应用技术的培养。

本套丛书的出版是在配合计算机应用型人才专业能力的培养和落实方面的初步尝试，在教材组织和编写上还会有许多不足和缺陷，需要进一步完善，我们衷心希望本套教材的出版能起到抛砖引玉的作用，也希望广大教育工作者加入到计算机应用型人才能力培养的研究和实践中来，并对相关的教材建设提出自己的宝贵意见。

丛书主编

陈明

# 丛书编委会



主任：陈明

副主任：王锁柱

委员：(以姓氏拼音为序)

曹永存	郝莹	贾宗璞	解凯
刘贵龙	孟庆昌	任化敏	袁薇
张建林	张晓明	周苏	

联络人：姚蕾

# 前 言

Java EE 技术经过多年的发展日趋成熟,已经成为最佳的企业应用解决方案之一。Java EE 是 Java 的高级应用部分,它与市场紧密衔接,容易使学生掌握前沿应用技术,理解和掌握新的软件开发思想,有利于培养学生面向实际的综合应用开发能力。为了适应形势的发展,许多高校开始开设 Java EE 课程。但目前市场上有关 Java EE 的书多为技术参考书,内容偏多、偏难,不能满足学生学习的需要。经过一段时间的思考和研究,并基于多年的教学及 Java 开发经验,我们编写了本教材。

鉴于 Java EE 的内容较多,教材的编写有一定难度,在编写过程中,我们主要注意了以下几点:

(1) 贯彻项目驱动、设计主导、案例教学的思想。作为实践性很强的课程,其主要目的是培养学生的实战能力,因此本教材注重贯彻项目驱动、设计主导、案例教学的思想。前七章以网络书城项目为线索,根据项目需要逐步引入框架,其中第 1 章就讲解了分层架构,并给出了网络书城项目的设计,后续章节主要围绕书城案例讲解知识,并将案例分解到知识当中,使案例与知识有机结合。最后,第 8 章给出基于新技术的软件论坛系统完整的设计与实现,使设计清晰化,并达到较高的水平。

(2) 基于需要选择内容,突出实用,讲解精练。本教材在保证知识系统性的同时,注重精选内容。从典型的项目出发,基于项目需要展开知识,围绕知识讲解案例,对知识的讲解不求全而细,而是强调实用,突出重点。在案例选择上不追求复杂化,而是力求典型性。案例之间相互联系,案例与知识相辅相成,形成有机的整体,使知识讲解直接化,既有利于学生学习知识,又有利于指导学生实践。

(3) 跟踪 Java 新发展,注意适应市场需求,及时引进新内容。本教材选择目前最新版的 NetBeans 6.9.1 开发平台和 GlassFish 3.0.1 服务器,不仅易学易用,而且便于跟踪新技术。在各部分知识的讲解中,力求适应新发展,例如 Struts2 选择了最新版的 Struts 2.8.1,持久化技术重点介绍 JPA,JSF 部分选择 JSF2 等。通过本教材不仅可以学习 Struts+Spring+Hibernate JPA 架构,还可以学习 EJB+Hibernate JPA+JSF 架构。

本书共 8 章,每一章的主要内容与课堂教学的建议学时安排为:

第 1 章主要内容包括 Java EE 简介,搭建 Java EE 开发环境,Java EE 应用中的分层架构。通过本章,主要学习如何搭建开发环境,如何构建企业级应用程序,建议使用 4 学时。

第 2 章主要内容包括 JDBC 技术,Servlet 技术,过滤器和监听器,JSP、标签和表达式语言。通过本章,主要学习 Java Web 应用开发的基本技术,力求精练,不要求过细,建议使用 10 学时。

第 3 章主要内容包括 Struts2 概述,Action 与结果类型,输入验证,Struts2 国际化、标签库和 OGNL 表达式,拦截器、文件上传和下载。通过本章,主要学习控制层实现技术,建议使用 12 学时。



第4章主要内容包括 JPA 概述, 实体对象映射, 实体关系映射, 实体操作与事务管理。通过本章, 主要学习持久化层实现技术, 建议使用 12 学时。

第5章主要内容包括 Spring 基础, Spring 的 AOP, Spring 集成其他框架, Spring 事务管理与任务调度。通过本章, 主要学习轻量级架构技术, 涉及系统整合、事务处理、任务调度等, 建议使用 10 学时。

第6章主要内容包括 EJB 概述, 无状态会话 Bean, 有状态会话 Bean, 单例 Bean、计时器 Bean 和消息驱动 Bean, EJB 中的拦截器和安全域。通过本章, 主要学习重量级架构技术, 建议使用 10 学时。

第7章主要内容包括 JSF 概述, JSF 表达式语言与标签, JSF 数据传递方法, 数据类型转换与验证, 消息处理与国际化支持, JSF 生命周期与事件处理。通过本章, 主要学习与 EJB 结合较多的新的标准 Java 架构技术, 建议使用 8 学时。

第8章主要内容包括系统分析与设计, 数据访问层实现, 业务逻辑层实现, 系统配置, 控制层实现, 视图层设计。通过本章, 主要通过论坛系统学习基于 EJB 架构的应用设计, 建议使用 6 学时。

如果课程学时为 48, 建议选择前五章进行教学; 如果课程学时为 72, 建议选择全书八章进行教学。

由于时间仓促、作者水平有限、书中难免存在疏漏和不足, 恳请读者批评指正, 使本书得以改进和完善。

编者

2010年10月于北京

# 目 录

出版者的话  
丛书序言  
丛书编委会  
前言

第 1 章 Java EE 概述 .....	1	2.1.4 JDBC 数据源与连接池 .....	33
1.1 Java EE 简介 .....	1	2.2 Servlet 技术 .....	36
1.1.1 Java EE 框架的产生 .....	1	2.2.1 什么是 Servlet .....	36
1.1.2 什么是 Java EE .....	2	2.2.2 如何建立 Servlet .....	36
1.1.3 Java EE 规范新概念 .....	3	2.2.3 案例 2-2 建立超级管理员 登录 Servlet .....	37
1.1.4 Java EE 常用技术 .....	4	2.2.4 Servlet 的配置 .....	40
1.2 搭建 Java EE 开发环境 .....	5	2.2.5 转发和重定向 .....	40
1.2.1 安装 JSEDK .....	5	2.2.6 处理请求和响应 .....	41
1.2.2 安装 Java EE 开发工具 .....	6	2.2.7 案例 2-3 使用 Servlet 实现 验证码 .....	43
1.2.3 安装 MySQL .....	8	2.3 过滤器和监听器 .....	45
1.2.4 安装 GlassFish .....	13	2.3.1 过滤器 Filter .....	45
1.3 Java EE 应用中的分层架构 .....	15	2.3.2 案例 2-4 设计一个编码 过滤器 .....	45
1.3.1 分层架构模式 .....	16	2.3.3 监听器 Listener .....	47
1.3.2 Java EE 中的三层结构 .....	17	2.3.4 Servlet 上下文与会话 .....	48
1.3.3 结合 MVC 模式的分层 结构 .....	17	2.3.5 案例 2-5 统计访问人数和 在线人数 .....	49
1.3.4 案例 1-1 网络书城系统 分析与设计 .....	17	2.4 JSP、标签和表达式语言 .....	51
本章小结 .....	27	2.4.1 JSP 概述 .....	51
习题 .....	27	2.4.2 JSP 隐含对象 .....	52
第 2 章 Java Web 应用开发基础 .....	28	2.4.3 自定义标签的设计与 使用 .....	53
2.1 JDBC 技术 .....	28	2.4.4 案例 2-6 设计图书分类 标签 .....	53
2.1.1 JDBC 简介 .....	28	2.4.5 表达式语言 EL 和标准 标签库 JSTL .....	56
2.1.2 JDBC 访问数据库的基本 步骤 .....	29	2.4.6 案例 2-7 实现图书分类的 管理 .....	60
2.1.3 案例 2-1 网络书城系统 数据访问层的实现 .....	29	本章小结 .....	64

习题 .....	64	4.1.5 案例 4-1 在书城项目中 配置 Hibernate JPA .....	107
<b>第 3 章 MVC 框架技术 Struts2</b> .....	65	<b>4.2 实体对象映射</b> .....	108
3.1 Struts2 概述 .....	65	4.2.1 实体映射基础 .....	108
3.1.1 Struts 框架的历史 .....	65	4.2.2 主键映射 .....	110
3.1.2 Struts2 下载与配置 .....	66	4.2.3 映射特殊类型 .....	111
3.1.3 案例 3-1 使用 Struts2 实现 客户登录 .....	67	4.2.4 案例 4-2 书城项目中客户 实体类的设计 .....	112
3.1.4 Struts2 的 MVC 原理 .....	70	<b>4.3 实体关系映射</b> .....	115
3.1.5 Struts2 组件 .....	71	4.3.1 实体关系概述 .....	115
3.1.6 Struts2 的中文乱码问题 .....	73	4.3.2 一对一映射 .....	115
3.2 Action 与结果类型 .....	73	4.3.3 一对多映射 .....	118
3.2.1 Action 的实现类 .....	73	4.3.4 多对一映射 .....	119
3.2.2 在 Action 中访问 Web 对象 .....	76	4.3.5 案例 4-3 书城项目的其他 实体类设计 .....	121
3.2.3 多方法的 Action .....	76	<b>4.4 实体操作与事务管理</b> .....	123
3.2.4 result 类型 .....	77	4.4.1 JPA 实体操作 .....	123
3.3 输入验证 .....	79	4.4.2 JPA 查询 .....	125
3.3.1 使用 validate() 方法 .....	79	4.4.3 JPA 事务管理 .....	128
3.3.2 使用验证文件验证 .....	80	4.4.4 辅助类设计 .....	129
3.3.3 案例 3-2 实现客户登录和 注册验证(使用配置文件 验证) .....	82	4.4.5 案例 4-4 基于 JPA 的书城 Dao 设计 .....	130
3.4 Struts2 国际化、标签库和 OGNL 表达式 .....	86	本章小结 .....	133
3.4.1 Struts2 国际化 .....	86	习题 .....	133
3.4.2 常用 Struts2 标签 .....	87	<b>第 5 章 轻量级框架 Spring</b> .....	135
3.4.3 Struts2 的 OGNL .....	90	5.1 Spring 基础 .....	135
3.5 拦截器、文件上传和下载 .....	91	5.1.1 Spring 框架 .....	135
3.5.1 拦截器 .....	91	5.1.2 Spring 控制反转 IoC .....	136
3.5.2 文件上传和下载 .....	93	5.1.3 Bean 的配置与 BeanFactory .....	139
3.5.3 案例 3-3 实现图书的 添加 .....	95	5.1.4 案例 5-1 在书城项目中 使用 Spring .....	140
本章小结 .....	99	<b>5.2 Spring 的 AOP</b> .....	144
习题 .....	100	5.2.1 AOP 的概念 .....	144
<b>第 4 章 持久化技术 JPA</b> .....	101	5.2.2 使用 Spring 中的 AOP .....	146
4.1 JPA 概述 .....	101	5.2.3 基于 XML Schema 和 Annotation 的 Spring AOP 配置 .....	150
4.1.1 JPA 及其优势 .....	101	5.2.4 案例 5-2 书城项目中的 日志实现 .....	151
4.1.2 Hibernate JPA .....	102	<b>5.3 Spring 集成其他框架</b> .....	152
4.1.3 实体与实体管理器 .....	103	5.3.1 Spring 与 Struts2 集成 .....	152
4.1.4 持久性单元 .....	106		

5.3.2	Spring 与 JPA 集成	152
5.3.3	SSHJ 架构	156
5.3.4	案例 5-3 基于 SSHJ 架构 的书城项目配置	156
5.4	Spring 事务管理与任务 调度	157
5.4.1	事务的基本配置	157
5.4.2	事务的传播属性	158
5.4.3	使用 aop 及 tx 标签配置 事务	160
5.4.4	使用标注配置事务	160
5.4.5	任务调度	161
5.4.6	案例 5-4 书城项目中 事务和任务调度的配置	162
	本章小结	164
	习题	165
第 6 章	EJB 架构	166
6.1	EJB 概述	166
6.1.1	什么是 EJB	166
6.1.2	EJB 组件类型	167
6.1.3	EJB 中的依赖注入	168
6.1.4	案例 6-1 使用 NetBeans 创建企业级书城项目	169
6.2	无状态会话 Bean	171
6.2.1	创建无状态会话 Bean	171
6.2.2	访问无状态会话 Bean	172
6.2.3	案例 6-2 创建书城的 无状态会话 Bean	173
6.3	有状态会话 Bean	176
6.3.1	创建有状态会话 Bean	176
6.3.2	会话 Bean 的生命周期	176
6.3.3	案例 6-3 书城的购物车 会话 Bean	178
6.4	单例会话 Bean、计时器 Bean 和消息驱动 Bean	180
6.4.1	单例会话 Bean	180
6.4.2	计时器 Bean	182
6.4.3	JMS 与消息驱动 Bean	183
6.4.4	案例 6-4 书城项目中的 订单消息驱动 Bean	187
6.5	EJB 中的拦截器和安全域	190
6.5.1	EJB 中的拦截器	190
6.5.2	验证、授权与登录验证	193
6.5.3	案例 6-5 基于安全域实现 书城项目的管理员登录 验证	194
	本章小结	198
	习题	198
第 7 章	JavaServer Faces 技术	199
7.1	JSF 概述	199
7.1.1	什么是 JSF	199
7.1.2	JSF 的优势	200
7.1.3	JSF 应用的组成	200
7.1.4	模板页面与适配页面	205
7.1.5	案例 7-1 在书城项目中 使用 JSF	207
7.2	JSF 表达式语言与标签	211
7.2.1	JSF 表达式语言	211
7.2.2	JSF 的 HTML 标签	212
7.2.3	JSF 核心标签	216
7.2.4	案例 7-2 书城客户注册 功能实现	217
7.3	JSF 数据传递方法	218
7.3.1	从 JSF 页面传递参数给 受管 Bean	218
7.3.2	在请求之间传递对象	219
7.3.3	在受管 Bean 之间通信	220
7.4	数据类型转换与验证	220
7.4.1	标准转换器	220
7.4.2	自定义转换器	221
7.4.3	标准验证器	223
7.4.4	自定义验证器	224
7.4.5	案例 7-3 为客户注册页 增加验证功能	225
7.5	消息处理与国际化支持	227
7.5.1	消息处理	227
7.5.2	国际化支持	228
7.6	JSF 生命周期与事件处理	231
7.6.1	JSF 生命周期	231
7.6.2	Phase 事件与 JSF 监听器	232
7.6.3	动作事件	233
7.6.4	即时事件	233
7.6.5	值变事件	234
	本章小结	235

习题	236	8.3.5 BbsMessageService 实现	252
<b>第 8 章 基于 EJB 架构的论坛系统</b>		8.4 系统配置	253
设计	237	8.4.1 连接池和数据源配置	253
8.1 系统分析与设计	237	8.4.2 持久化单元配置	253
8.1.1 系统分析	237	8.4.3 引入 JSF 框架	253
8.1.2 总体设计	237	8.4.4 安全验证配置	254
8.1.3 实体类(数据模型)设计	240	8.5 控制层实现	257
8.1.4 接口设计	244	8.5.1 控制层基类	257
8.2 数据访问层实现	246	8.5.2 版块控制类	259
8.2.1 BaseDao 实现	246	8.5.3 主题控制类	260
8.2.2 BbsUserDao 实现	248	8.5.4 帖子控制类	263
8.2.3 BbsLoginDao 实现	248	8.5.5 用户控制类	265
8.2.4 BbsForumDao 实现	249	8.6 视图层设计	268
8.2.5 BbsThemeDao 实现	249	8.6.1 界面的总体设计	268
8.2.6 BbsMessageDao 实现	249	8.6.2 主题视图设计	272
8.3 业务逻辑层实现	250	8.6.3 帖子视图设计	276
8.3.1 BbsUserService 实现	250	8.6.4 用户视图设计	280
8.3.2 BbsLoginService 实现	250	本章小结	284
8.3.3 BbsForumService 实现	251	习题	284
8.3.4 BbsThemeService 实现	251		

# Java EE 概述

## 本章内容

- Java EE 简介
- 搭建 Java EE 开发环境
- Java EE 应用中的分层架构

Java EE(Java Enterprise Edition)是建立在 Java 平台上的企业级应用的解决方案。Java EE 基于 Java SE(Java Standard Edition)平台,提供了一组用于开发和运行的可移植的、健壮的、可伸缩的、可靠的和安全的服务器端应用程序的 API(Application Programming Interface, 应用程序编程接口)。

## 1.1 Java EE 简介

### 1.1.1 Java EE 框架的产生

Sun 公司在 1998 年发表 JDK1.2 版本的时候,开始使用名称 Java 2 Platform,即 Java 2 平台,修改后的 JDK 称为 Java 2 Platform Software Developing Kit,即 J2SDK,并分为标准版(Standard Edition, J2SE)、企业版(Enterprise Edition, J2EE)和微型版(Micro Edition, J2ME)。2006 年 5 月, Sun 公司推出 Java SE 5,此时 Java 的各种版本又进行了更名, J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

J2EE 是纯粹基于 Java 的解决方案。1998 年, Sun 发布了 EJB 1.0 标准。EJB 为企业级应用中的数据封装、事务处理、交易控制等功能提供良好的技术基础。至此, J2EE 平台的三大核心技术 Servlet、JSP 和 EJB 都已先后问世。1999 年, Sun 正式发布了 J2EE 的第一个版本。紧接着,遵循 J2EE 标准、为企业级应用提供支撑平台的各类应用服务软件相继涌现出来。IBM 的 WebSphere、BEA 的 WebLogic 都是这一领域里成功的商业软件平台。随着开源运动的兴起, JBoss 等开源的应用服务器软件新秀也吸引了许多用户的注意力。2003 年, Sun 的 J2EE 版本已经升级到 1.4 版,其中 3 个关键组件的版本也升级到了 Servlet 2.4、JSP 2.0 和 EJB 2.1。至此, J2EE 体系及相关的软件产品已经成为 Web 服务端开发的一个强有力的支撑环境。

但从 1999 年诞生的第一个 J2EE 版本一直到 J2EE 1.4 版本,总被人们抱怨。这并不是因为它不够强大,恰恰相反,正是因为它太强大了,强大到让人难以使用。实现一个简单的 J2EE 程序,就需要大量的配置文件,尽管有些配置文件不是必需的。Sun 公司一直在试图改变这一切,但一直未能如愿。2002 年, J2EE 1.4 推出后, J2EE 的复杂程度达到了顶点。尤其是 EJB 2.0,开发和调试的难度非常大。也许是要下决心改变这一切,或者是受到市场和开发人员的压力, Sun 终于在 2006 年 5 月正式发布了 J2EE 1.5(现改名为 Java EE 5)规范,并宣称 Java EE

5 将是 Java EE 史上最简单的版本, 将大大降低开发难度。目前 Java EE 已经发展到 Java EE 6。图 1-1 说明了 Java EE 的发展过程。

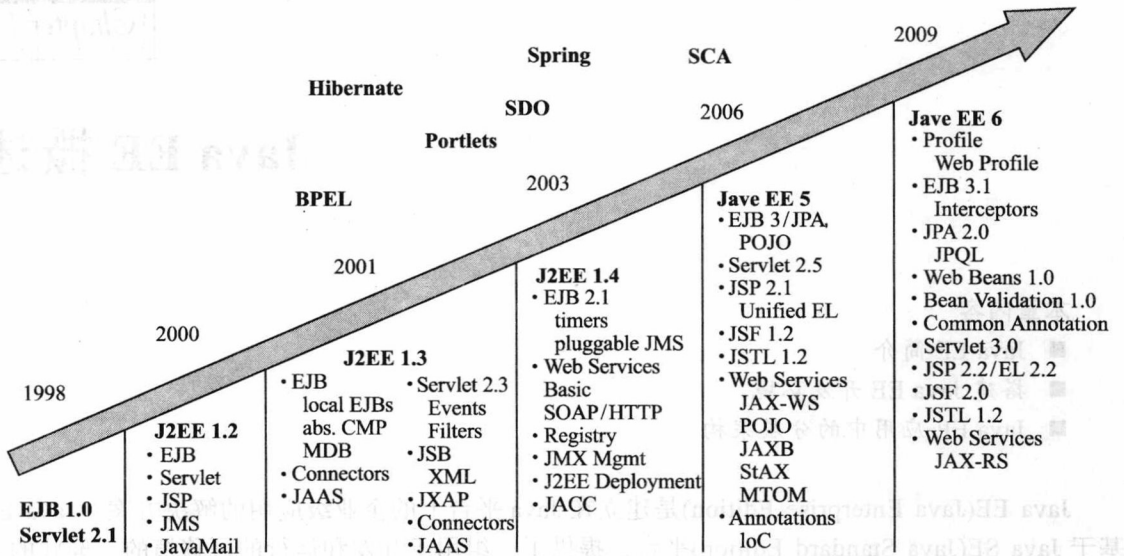


图 1-1 Java EE 的发展过程

## 1.1.2 什么是 Java EE

Java EE 是 Java 平台的企业版, 是一套技术架构。Java EE 可提高应用程序的可移植性、安全性与复用价值, 其核心是一组技术规范与指南。Java EE 的产生使开发人员只需要注重商业逻辑与架构设计。

Java EE 是 J2EE 版本的后续版本, 是 J2EE 技术的新生和发展。Java EE 技术具有 J2SE 平台的所有功能, 同时还提供对 EJB、Servlet、JSP、XML 等技术的全面支持。Java EE 的最终目标是成为一个支持企业级应用开发的体系结构, 简化企业解决方案的开发、部署和管理等复杂问题。事实上, Java EE 已经成为企业级开发的工业标准和首选平台。Java EE 是一个标准而不是一个产品, 各个平台开发商按照 Java EE 规范开发不同的 Java EE 应用服务器。推出 Java EE 框架的目的是为了克服传统 C/S 模式的弊端, 迎合 B/S 架构的潮流。

Java EE 不是 Java 标准版(Java SE)的替代品。Java SE 是 Java EE 的核心部分, 它为 Java EE 提供了基本的语言框架。正如后面将会看到的那样, Java EE 由很多层组成, 而 Java SE 只是 Java EE 所有组件的基础。作为 Java 开发人员, 可能已经学会使用 Swing 或者 AWT(Abstract Window Toolkit)的组件来建立用户界面。对于 Java EE 应用程序, 仍然可以用这些组件来建立用户界面。此外还可以开发基于 HTML 的用户界面。正因为 Java SE 是 Java EE 的核心, 所以过去所学的任何有关 Java 的知识仍然有用。

此外, Java EE 还有另一套用于创建用户界面的 API, 称为 JSF(JavaServer Faces), 这是 Java 最新的技术之一。Java EE 平台也支持开发应用程序的中间层, 包括业务逻辑和与后端数据源连接。可以利用熟悉的 J2SE 组件和 API, 结合 Java EE 的组件和 API 来构建这部分应用。

为了开发企业级的应用, Java EE 定义了大量的服务, 用于实现企业级的应用和大量的基础结构。编写具有可扩展性的、健壮的、安全的和易于维护的分布式应用程序, 需要用到大量的系统级功能。这些重要的基础结构模块包括了安全性、数据库访问和事务控制等系统级功能。其中, 安全性保证了用户只可以访问应用中那些被赋予相应权限的部分; 数据库访问

是最基本的组件之一，便于应用程序存储和读取数据；对事务的支持则保证在恰当的时间更新相应的数据。

Sun 公司的 Java EE 软件开发包(Software Development Kit, SDK)提供了一个参考实现，其中包括一个工作模型。因为这是 Sun 公司完全按照规范实现的，并且是免费的，所以本书也采用了它。Java EE 框架所支持的服务以及工作流程，如图 1-2 所示。

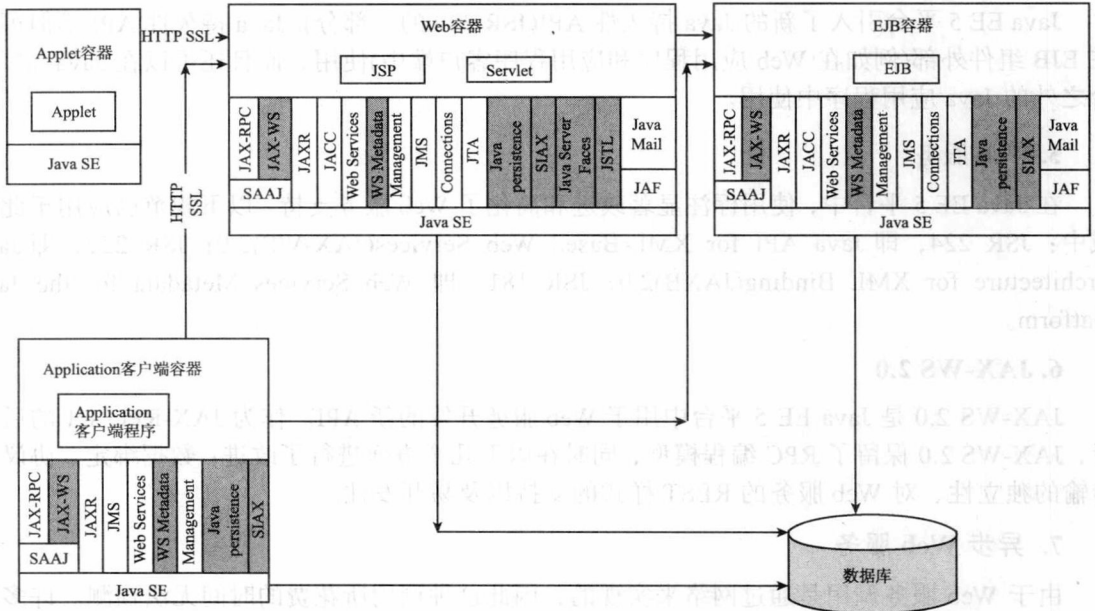


图 1-2 Java EE 框架图

图 1-2 中灰色部分是 Java EE 5 的新功能。可以看到，在 Web 层主要加入了 JSF 这个新的表示层框架，同时还引入了新的 Java 持久化标准。

需要指出的是，这个 Java 持久化标准可以嵌入 Web 层调用，所以它肯定不会从属于 EJB 标准。这样，当前所有的 Java 持久层标准，如 JDBC、JDO、Hibernate/Entity Bean 将可能统一，从而减少用户选择架构的烦恼。

从图 1-2 可以看出，无论 Web 结构还是 EJB 结构，实现的功能相差不多。这样业务核心组件就可以根据需要部署在 Web 或 EJB 容器中运行，而不依赖具体的 Java EE 容器。

### 1.1.3 Java EE 规范新概念

#### 1. 标注取代部署描述符

Java EE 5/6 采用标注来代替部署描述符，大大简化了 Java EE 程序的开发。标注(也称为注解、注释)是 Java 修饰符，与代码中指定的 public 和 private 类似。例如，EJB 3 规范为 Bean 类型、接口类型、资源引用、事务属性、安全性等定义了标注。JAX-WS 2.0 规范为 Web 服务提供了一组类似的标注。

#### 2. 简化的 EJB 软件开发

使用新的 EJB 3 API 可以减少开发者的工作量，从而使开发者可以更轻松地进行软件开发。换句话说，就是使用了更少的类和代码。这是因为容器承担了更多的工作。在可视化开发工具 IDE 中，可以辅助更加快捷地建立企业应用程序。



### 3. 使用依赖关系注入访问资源

通过注入依赖关系,对象可以使用标注直接请求外部资源。这样代码会更简洁,因为不再需要将代码与资源创建和查找的代码混杂在一起。可以在 EJB 组件、Web 容器和客户端中使用资源注入。例如,资源注入使用@Resource 标注,会话 Bean 的注入使用@EJB 标注。

### 4. Java 持久性 API 模型

Java EE 5 平台引入了新的 Java 持久性 API(JSR 220 的一部分)。Java 持久性 API 不但可以在 EJB 组件外部(例如在 Web 应用程序和应用程序客户端中)使用,而且还可以在 Java EE 平台之外的 Java 应用程序中使用。

### 5. Web 服务

在 Java EE 5 平台中,使用标注显著改进和简化了 Web 服务支持。以下规范已应用于此领域中:JSR 224,即 Java API for XML-Based Web Services(JAX-WS)2.0;JSR 222,即 Java Architecture for XML Binding(JAXB)2.0;JSR 181,即 Web Services Metadata for the Java Platform。

### 6. JAX-WS 2.0

JAX-WS 2.0 是 Java EE 5 平台中用于 Web 服务开发的新 API。作为 JAX-RPC 1.1 的后继者,JAX-WS 2.0 保留了 RPC 编程模型,同时在以下几个方面进行了改进:数据绑定、协议和传输的独立性、对 Web 服务的 REST 样式的支持以及易开发性。

### 7. 异步 Web 服务

由于 Web 服务调用是通过网络来实现的,因此这种调用所花费的时间无法预测。许多客户端因必须等待服务器的响应而严重影响了性能。为了避免出现性能降低的问题,JAX-WS 2.0 提供了新的异步客户端 API。使用此 API,应用程序编程人员无须自己创建线程,可以依赖于 JAX-WS 为他们管理长时间运行的远程调用。异步方法可以与 WSDL 生成的任何接口以及动态性更强的 Dispatch API 联合使用。

## 1.1.4 Java EE 常用技术

### 1. Servlet

Servlet 是 Java 平台上的 CGI 技术。Servlet 在服务器端运行,动态地生成 Web 页面。与传统的 CGI 和许多其他类似 CGI 的技术相比,Java Servlet 具有更高的效率并更容易使用。对于 Servlet,重复的请求不会导致同一程序的多次转载,它是依靠线程的方式来支持并发访问的。

### 2. JSP

JSP(Java Server Page)是一种实现普通静态 HTML 和动态页面输出混合编码的技术。从这一点来看,非常类似于 Microsoft ASP、PHP 等技术。借助形式上的内容和外观表现的分离,Web 页面制作的任务,可以比较方便地划分给页面设计人员和程序员,并方便地通过 JSP 来合成。在运行时,JSP 将会被首先转换成 Servlet,并以 Servlet 的形态编译运行,因此它的效率和功能与 Servlet 相比没有差别,一样具有很高的效率。

### 3. JDBC

JDBC(Java Database Connectivity,数据库访问接口),它使数据库开发人员能够用标准 Java API 编写数据库应用程序。JDBC API 主要用来连接数据库和直接调用 SQL 命令执行各种 SQL 语句。利用 JDBC API 可以执行一般的 SQL 语句、动态 SQL 语句及带 IN 和 OUT 参数的存储