

 高等教育规划教材

Java EE 框架开发技术与案例教程

张继军 董卫 编著



免费提供电子教案、案例源代码

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS



高等教育规划教材

Java EE 框架开发技术与案例教程

张继军 董卫 编著

机械工业出版社

本书内容包括: Java Web 开发的基础技术(HTML、JavaScript、CSS、JSP、Servlet、过滤器、监听器、JavaBean 和 JDBC 等)、Struts 2 框架技术及其相关的应用案例、Hibernate 框架技术及其应用案例、Spring 框架及其子框架技术(AOP 编程技术、Spring 的数据库访问框架和 Spring 的 Web 框架)与应用案例,以及 Spring 与 Struts 2、Hibernate 整合与应用案例等有关内容。

本书提供了丰富的案例,通过这些具体完整的应用案例,对开发、集成、部署,以及具体实现的过程和方法给出了详细介绍和说明,包括如何进行设计、运行和测试,力求让读者通过这些案例领会并掌握 Java EE 及 Struts 2、Hibernate、Spring 框架的开发技巧、设计方法,以及它们的整合应用。

本书主要面向初学者,特别适合高等院校和职业院校的学生作为学习 Java EE 框架技术开发应用程序相关课程的教材,也可作为基于 Java EE 框架技术开发应用系统的开发人员的学习资料或参考书。

本书配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册,审核通过后下载,或联系编辑索取(QQ: 2850823885, 电话: 010-88379739)。

图书在版编目(CIP)数据

Java EE 框架开发技术与案例教程 / 张继军, 董卫编著. —北京: 机械工业出版社, 2016.4

高等教育规划教材

ISBN 978-7-111-54186-8

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

中国版本图书馆 CIP 数据核字(2016)第 171991 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: 郝建伟 责任校对: 张艳霞

责任印制: 李 飞

北京铭成印刷有限公司印刷

2016 年 9 月第 1 版·第 1 次印刷

184mm×260mm·20.75 印张·512 千字

0001—3000 册

标准书号: ISBN 978-7-111-54186-8

定价: 49.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

服务咨询热线: (010) 88379833

机工官网: www.cmpbook.com

读者购书热线: (010) 88379649

机工官博: weibo.com/cmp1952

教育服务网: www.cmpedu.com

封面无防伪标均为盗版

金 书 网: www.golden-book.com

出版说明

当前,我国正处在加快转变经济发展方式、推动产业转型升级的关键时期。为经济转型升级提供高层次人才,是高等院校最重要的历史使命和战略任务之一。高等教育要培养基础性、学术型人才,但更重要的是加大力度培养多规格、多样化的应用型、复合型人才。

为顺应高等教育迅猛发展的趋势,配合高等院校的教学改革,满足高质量高校教材的迫切需求,机械工业出版社邀请了全国多所高等院校的专家、一线教师及教务部门,通过充分的调研和讨论,针对相关课程的特点,总结教学中的实践经验,组织出版了这套“高等教育规划教材”。

本套教材具有以下特点:

- 1) 符合高等院校各专业人才的培养目标及课程体系的设置,注重培养学生的应用能力,加大案例篇幅或实训内容,强调知识、能力与素质的综合训练。
- 2) 针对多数学生的学习特点,采用通俗易懂的方法讲解知识,逻辑性强、层次分明、叙述准确而精炼、图文并茂,使学生可以快速掌握,学以致用。
- 3) 凝结一线骨干教师的课程改革和教学研究成果,融合先进的教学理念,在教学内容和方法上做出创新。
- 4) 为了体现建设“立体化”精品教材的宗旨,本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、源代码或源程序、教学大纲、课程设计和毕业设计指导等资源。
- 5) 注重教材的实用性、通用性,适合各类高等院校、高等职业学校及相关院校的教学,也可作为各类培训班教材和自学用书。

欢迎教育界的专家和老师提出宝贵的意见和建议。衷心感谢广大教育工作者和读者的支持与帮助!

机械工业出版社

前 言

Java EE 是目前应用相当广泛的一种软件开发技术,已经成为主要的企业应用解决方案之一,它与市场、企业需求紧密联系,并且不断改进升级,将新思想、新技术、新方法融入其中,学习该技术能够掌握前沿技术,理解和掌握新的软件开发思想,有利于培养读者面向实际的综合开发能力和应用能力。目前,无论是高校的计算机专业、计算机相关专业,还是 IT 培训机构,都将其作为教学的重要内容,也是从事软件开发人员应当掌握的一门专业技术。

在 Java EE 平台上,整合应用 Spring、Struts 2、Hibernate 三大开发框架技术是目前最主流、最普遍使用的应用开发方式。Spring 框架可以支持多种 Java EE 技术和框架,使得 Java EE 应用的开发变得更简单、更有效;Struts 2 是基于 MVC 设计模式的、开源的 Java EE 编程框架,已得到了广泛应用,已经成为主流的 Web 编程框架;Hibernate 基于 ORM 实现了对数据库的访问与管理,已经成为访问数据库的主要框架技术。

针对这些框架技术,应该有一本能够简明介绍它们的原理、技术、应用,以及它们的整合应用的书籍。这样可以大大节约读者,特别是初学者的学习时间,提高学习质量和学习效果,真正掌握这些框架的使用与应用,为此,编者编写了本书。

本书的编写特别突出了三点:提高学生系统认知能力和系统设计能力的思想;贯穿项目驱动、设计主导、案例教学的思想;及时引进新思想、新方法、新内容,使读者适应社会需求和市场需求。

1) 突出“系统观点和系统设计”的思想:Java EE 应用的开发实际上是一个应用系统的开发,需要读者有一个整体的系统观念来组织、理解各部分的功能及其所使用的技术,在内容组织上围绕着提高“系统能力”,以提高读者的“系统设计能力”为目标。

2) 贯穿“项目驱动、设计主导、案例教学”的思想:通过典型的案例,将知识要点融入案例中,在求解案例时,利用有关的技术知识,使技术方法和知识与案例相辅相成;同时,每个案例都是一个 Java EE 应用系统,在设计中采用工程、系统的思想与方法。

3) Java EE 及其相关的框架技术是不断改进升级的,在本书的编写过程中,将其新思想、新技术、新方法融入其中,使读者能够掌握前沿应用技术,理解和掌握新的软件开发思想、技术与方法,有利于培养读者适应社会需求和市场需求的能力。本书采用了 Java EE 的新规范(Java EE 7),以及三大框架的新版本(Spring 4.0、Hibernate 4.0、Struts 2.3.24)。

书中的每个案例都按软件工程的思路给出了详细的设计思想、设计方法、实现步骤的分析和描述,使读者在阅读学习中逐渐培养应用系统的开发方法和技能,提高读者的设计能力,这也是本书不同于其他同类书的突出特点。

本书的编写是按 Java EE 应用所需技术的次序设置章节的,每种开发技术都提供了与其相关的开发案例。对每种技术,采用“技术的基本知识”→“技术的应用案例”→“使用该技术所遇到的问题及其解决方法”的线路组织内容,在应用中提出问题,解决问题,引导读者探讨并解决问题,提高读者的学习兴趣和积极性。

本书从实用的角度出发，同时为了利于读者学习，通过案例由浅入深地介绍了 Java EE 及 Struts 2、Hibernate、Spring 框架的基本原理、技术、使用方法和开发应用，以及它们的整合应用。

全书共 12 章，第 1~2 章介绍了 Java EE 应用开发所必需的开发环境和最基本的开发技术，主要包括：Java EE 开发环境的搭建、视图设计技术（HTML、JavaScript、CSS 和 JSP 等）、控制器技术（Servlet、过滤器及监听器等）、JavaBean 技术、JDBC 技术、MVC+DAO 模式的应用开发；第 3~5 章介绍了 Struts 2 框架技术，重点给出了该框架核心组件的使用与应用技巧，特别是第 5 章，基于软件工程的思想方法，给出了较完成的应用开发案例，引导读者理解和掌握应用程序的开发过程。第 6~7 章介绍了 Hibernate 框架技术，重点给出了该框架的使用方法，基于 Hibernate 的通用访问数据库的工具类，并对实体类之间的各类关系映射通过案例给出了较详细的介绍，进一步给出 Hibernate 事务管理、缓存管理的方法与处理；第 8~11 章介绍了 Spring 框架技术，重点介绍了 Spring IoC、AOP 编程技术、Spring 的数据库访问框架、Spring 的 Web 框架，以及 Spring 与 Struts 2 整合应用，Spring 与 Hibernate 整合应用；第 12 章给出了整合 Spring+ Struts 2+Hibernate 的综合应用案例。

本书中所介绍的案例和例题都是在 Windows 7、Eclipse Java EE 4.2、MySQL 数据库环境下进行的，使用的框架为 Spring 4.0、Hibernate 4.0 和 Struts 2.3.24 给出的开发设计，并调试运行通过。每个案例都按软件工程的思想，给出了完整的设计思想和设计步骤，以帮助读者顺利地完成任务。从应用程序的设计到应用程序的发布，读者都可以按照书中所讲述的内容实施。作为教材，每章后面都附有习题。

本书主要面向初学者，特别适合高等院校和职业院校的学生作为学习 Java EE 框架技术开发应用程序相关课程的教材，也可作为基于 Java EE 框架技术开发应用系统的开发人员的学习资料或参考书。

为了方便教师备课和学生的学习，本书还配有电子教案（PPT 文件）和案例的源代码。如有需要，可在机械工业出版社网站 www.cmpedu.com 下载。

感谢读者选择使用本书，由于时间仓促，加之作者水平有限，书中难免存在不妥之处，敬请广大读者原谅，并欢迎对本书内容提出意见和建议，不胜感激。

编者

目 录

出版说明

前言

第 1 章 Java EE 概述	1
1.1 Java EE 简介	1
1.1.1 Java EE 组件与应用模型	1
1.1.2 Java EE API 及主要技术	2
1.2 Java EE 开发环境与开发工具	5
1.2.1 Java JDK 的下载与安装	6
1.2.2 Tomcat 服务器的安装与配置	7
1.2.3 Eclipse 集成开发工具的下载、安装与使用	8
本章小结	14
习题	14
第 2 章 Java EE Web 应用开发基础技术	15
2.1 表示层（视图层）技术	15
2.1.1 HTML 技术与应用案例	15
2.1.2 CSS 技术与应用案例	17
2.1.3 JavaScript 技术与应用案例	21
2.1.4 JSP 技术与应用案例	24
2.1.5 EL 表达式与应用案例	30
2.2 控制层技术	32
2.2.1 Servlet 技术与应用案例	32
2.2.2 过滤器技术与应用案例	34
2.2.3 监听器技术与应用案例	36
2.3 业务逻辑（模型）层技术	39
2.4 持久层（JDBC）技术	40
2.5 MVC+DAO 设计模式与应用案例	42
2.5.1 基于 MVC 模式算术运算的 Web 程序设计	43
2.5.2 基于 MVC 模式用户注册与登录系统的开发	45
本章小结	48
习题	48
第 3 章 Struts 2 框架基础	49
3.1 Struts 2 简介	49
3.2 搭建 Struts 2 开发环境	51

3.3	Struts 2 入门案例	52
3.3.1	入门案例 1——基于 Struts 2 计算任意两个数据的和	53
3.3.2	入门案例 2——基于 Struts 2 实现用户注册与登录	55
3.4	Struts 2 的配置	62
3.4.1	web.xml 配置文件	62
3.4.2	struts-default.xml 配置文件	62
3.4.3	struts.xml 配置文件	62
3.5	Struts 2 的控制层及 Action 设计与配置	65
3.5.1	Action 类的实现与传值	66
3.5.2	基于 XML 配置文件的 Action 配置与访问	70
3.5.3	多方法的 Action 设计与配置访问	71
3.5.4	Action 访问 Web 资源	75
3.5.5	基于注解的 Action 配置	79
3.6	案例——基于注解配置 Action 实现用户注册与登录系统	80
	本章小结	81
	习题	81
第 4 章	Struts 2 框架的视图与拦截器技术	82
4.1	Struts 2 的值栈及访问	82
4.2	Struts 2 的 OGNL 表达式与应用案例	84
4.3	Struts 2 常用标签	88
4.3.1	Struts 2 的通用标签与应用案例	88
4.3.2	Struts 2 的控制标签与应用案例	92
4.3.3	Struts 2 的表单标签与应用案例	94
4.4	Struts 2 的拦截器	97
4.4.1	Struts 2 拦截器的自定义实现	97
4.4.2	案例——文字过滤器的设计与应用	99
4.5	Struts 2 的国际化与应用案例	101
4.5.1	资源文件	102
4.5.2	资源文件的访问	102
4.5.3	Struts 2 的国际化应用案例	104
4.6	Struts 2 数据输入验证	105
4.6.1	Struts 2 的声明式输入验证	106
4.6.2	Struts 2 声明式输入验证案例——实现客户注册输入验证	110
4.6.3	Struts 2 的编程验证	113
4.6.4	Struts 2 编程验证案例——实现客户注册输入验证	114
4.7	Struts 2 文件上传及下载	114

4.7.1	文件上传与应用案例	115
4.7.2	文件下载与应用案例	117
	本章小结	119
	习题	119
第 5 章	基于 Struts 2 的名片管理系统	120
5.1	项目的需求说明	120
5.2	系统分析与设计	120
5.3	创建工程并搭建开发环境	124
5.4	数据库与数据表的设计与实现	126
5.5	访问数据库的通用工具类的设计与实现	126
5.5.1	数据库连接属性文件——c3p0.properties	127
5.5.2	数据库连接工具类——JdbcPoolUtils	127
5.5.3	数据库访问接口——DAO 接口	128
5.6	用户注册与登录模块的开发	128
5.6.1	用户模型类的实现	129
5.6.2	访问用户数据表 DAO 的实现类	129
5.6.3	用户 Action 类的实现与基于 XML 的 Action 的配置与验证	130
5.6.4	用户注册、登录管理视图的设计与实现	132
5.7	名片管理模块的开发	133
5.7.1	名片模型类的实现	134
5.7.2	访问名片数据表 DAO 的实现	134
5.7.3	名片管理 Action 类的实现与基于注释的 Action 的配置	138
5.7.4	名片管理的视图设计与实现	139
5.7.5	Excel 与数据库之间的数据交互及上传、下载的实现	143
5.7.6	名片回收站管理模块的设计与实现	146
5.8	讨论与思考	147
	本章小结	147
	习题	147
第 6 章	Hibernate 框架基础	148
6.1	Hibernate 概述	148
6.1.1	数据持久化与 ORM	148
6.1.2	Hibernate 框架的结构体系	149
6.1.3	Hibernate 的核心组件	149
6.1.4	Hibernate 运行过程	150
6.2	搭建 Hibernate 开发环境	151
6.3	Hibernate 入门案例——数据库中添加记录	153

6.4	Hibernate 的核心组件及其应用	155
6.4.1	Hibernate 配置文件	156
6.4.2	Hibernate 的 PO 对象	157
6.4.3	Hibernate 映射文件	158
6.4.4	Hibernate 核心类（接口）与应用	160
6.5	HQL 语言与 Query 接口及应用	163
6.5.1	HQL 语言	163
6.5.2	Query 接口与应用	166
6.6	Hibernate 操作辅助类的设计	169
6.7	Struts 2 与 Hibernate 整合的名片管理系统开发案例	173
	本章小结	177
	习题	177
第 7 章	Hibernate 框架的高级特性	178
7.1	Hibernate 的 QBC 查询	178
7.2	Native SQL	180
7.3	Hibernate 实体关联关系映射	182
7.3.1	一对一映射	182
7.3.2	一对多映射和多对一映射	186
7.3.3	多对多映射	191
7.4	Hibernate 组件映射	194
7.5	Hibernate 继承映射	195
7.6	基于注解方式实现映射	199
7.6.1	基于注解方式映射的简单案例	199
7.6.2	基于注解的实体对象映射	200
7.6.3	基于注解的实体关联关系映射	202
7.6.4	Hibernate 的注解映射应用案例	204
7.7	Hibernate 的事务管理	207
7.8	Hibernate 的 Cache 管理	209
7.8.1	一级 Cache	209
7.8.2	二级 Cache	209
	本章小结	210
	习题	210
第 8 章	Spring 框架基础	211
8.1	Spring 概述	211
8.2	搭建 Spring 开发环境	212
8.2.1	下载 Spring	212

8.2.2	安装 Springsource-tool-suite 插件	213
8.3	Spring 框架的工作原理与应用案例	215
8.3.1	Spring 的控制反转	215
8.3.2	Spring 框架的工作原理	218
8.3.3	应用案例 1——采用 Java 工程实现 Hello World! 的显示	218
8.3.4	应用案例 2——采用 Web 工程实现 Hello World! 的显示	221
8.4	Spring IoC 容器及其实例化与使用	222
8.5	基于 XML 文件方式的 Bean 配置	223
8.6	Spring 表达式——SpEL	231
8.7	基于注释方式的 Bean 配置	233
8.7.1	使用注释定义 Bean	233
8.7.2	基于注释的 Bean 自动装配	235
8.7.3	配置类组件扫描路径	237
	本章小结	238
	习题	238
第 9 章	Spring AOP——面向切面编程	240
9.1	Spring AOP 的基本概念与入门案例	240
9.1.1	AOP 的基本概念和 workflow	240
9.1.2	Spring AOP 编程的环境搭建与入门案例	244
9.2	基于 XML 配置的 AOP 编程	249
9.2.1	Spring AOP 的通知与切入点	249
9.2.2	基于 XML 的切面配置	251
9.2.3	基于 XML 配置的 AOP 编程案例	252
9.3	基于 AspectJ 注解配置的 AOP 编程	255
9.3.1	在 Spring 中启用 AspectJ 注解支持	255
9.3.2	基于 AspectJ 注解声明切面	256
9.3.3	基于 AspectJ 注解声明通知	257
9.3.4	基于 AspectJ 注解 AOP 编写案例	259
	本章小结	261
	习题	261
第 10 章	Spring 的数据库访问框架	262
10.1	Spring 数据库访问框架概述	262
10.2	Spring 对 JDBC 的支持	264
10.2.1	Spring+JDBC 与 DataSource 的整合应用	264
10.2.2	Spring+JDBC 与 C3P0 的整合应用	268
10.2.3	Spring+JDBC 与数据库连接属性文件的整合应用	269

10.3	Spring 与 Hibernate 的配置整合	270
10.3.1	Spring 与 Hibernate 整合的基本思想与整合步骤	270
10.3.2	Hibernate 数据源的配置与注入	271
10.3.3	Hibernate 的 SessionFactory 实例的配置创建	271
10.4	Spring 中的事务管理	273
10.4.1	Spring 中的事务管理方法与事务管理器	273
10.4.2	Spring 编程式管理事务方式	274
10.4.3	基于 XML 的声明式管理事务	274
10.4.4	基于注解的声明式管理事务	275
10.4.5	事务属性的配置	276
10.5	Spring 与 Hibernate 整合的应用案例	279
10.5.1	基于 XML 配置的用户注册与登录的开发	279
10.5.2	基于注解配置的用户注册与登录的开发	283
	本章小结	285
	习题	285
第 11 章	Spring 对 Web 应用的支持	286
11.1	Web 应用访问 Spring	286
11.1.1	实例化 Spring IoC	286
11.1.2	在 JSP 中访问 Spring IoC 实例对象	287
11.1.3	在 Servlet 中访问 Spring IoC 实例对象	288
11.2	Spring 与 Struts 2 的集成	289
11.2.1	Spring 与 Struts 2 集成的方法与步骤	290
11.2.2	Spring 与 Struts 2 集成的配置	290
11.3	Spring 与 Struts 2 集成应用案例	293
11.3.1	基于注解声明的集成应用——人员信息的添加与删除	293
11.3.2	基于 XML 配置的集成应用——人员信息的添加与删除	297
	本章小结	298
	习题	298
第 12 章	Spring、Struts 2、Hibernate 整合与应用案例	299
12.1	Spring、Struts 2 和 Hibernate 框架整合与环境搭建	299
12.1.1	整合 jar 包	299
12.1.2	整合配置文件	300
12.2	案例——基于 SSH 整合的名片管理系统	304
12.2.1	项目需求说明	304
12.2.2	系统分析与设计	304
12.2.3	创建工程并搭建开发环境	308

12.2.4	数据库与数据表的设计与实现	308
12.2.5	属性文件与配置文件	308
12.3	访问数据库的通用接口及实现类	310
12.4	用户注册与登录子系统的实现	312
12.4.1	用户实体类的实现	312
12.4.2	访问用户数据表 DAO 的实现类	313
12.4.3	用户 Service 层的接口与实现类	313
12.4.4	用户 Action 类的实现与基于 XML 的 Action 的配置与验证	314
12.4.5	用户注册、登录管理视图的设计与实现	315
12.5	名片管理子系统的实现	315
12.5.1	名片实体类的实现	315
12.5.2	访问名片数据表 card 的 DAO 实现类	316
12.5.3	名片 Service 层的接口与实现类	316
12.5.4	名片 Action 类的实现与基于注解方式配置 Action	317
12.5.5	名片管理视图的设计与实现	318
12.6	其他相关模块的设计与实现	318
12.7	讨论与思考	318
	本章小结	318
	习题	319
	参考文献	320

第 1 章 Java EE 概述

Java EE (Java Enterprise Edition) 可被称为 Java 平台企业版。该平台为程序开发人员提供了一组强大的 API，从而缩短了开发周期，降低了应用复杂性，并改善了应用性能，从而使 Java 企业级应用开发变得简单、快捷。

本章简要介绍 Java EE 平台组成部分、Java EE 应用软件体系结构；基于 Java EE 平台开发 Web 应用程序所需要的主流技术和常用框架技术，以及开发 Java Web 应用所需要的开发环境、运行环境和开发工具。

1.1 Java EE 简介

Java EE 是由 J2EE 更名而来的。Sun 公司（已被甲骨文公司收购）在 1998 年发布 JDK1.2 版本的时候，开始使用 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 SE5，此时，Java 的各种版本依次更名为 Java EE、Java SE 和 Java ME。

随着 Java 技术的发展，Java EE 得到了迅速发展，如今，Java EE 不仅仅是指一种标准平台，更表达了一种软件架构和设计思想。目前的版本已经发展为 Java EE 7。本教材基于 Java EE 7 介绍其相关的设计思想、开发方法及开发技术。

本节将简单介绍 Java EE 平台，主要内容包括：Java EE 应用分层架构、分层模式及 Java EE 的结构等有关知识。

1.1.1 Java EE 组件与应用模型

Java EE 7 提供了一套完整、全面、集成的组件帮助用户构建企业和 Web 应用程序。图 1-1 给出了各组件之间的关系，各组件的功能与特点将在 1.1.2 节进行详细说明。

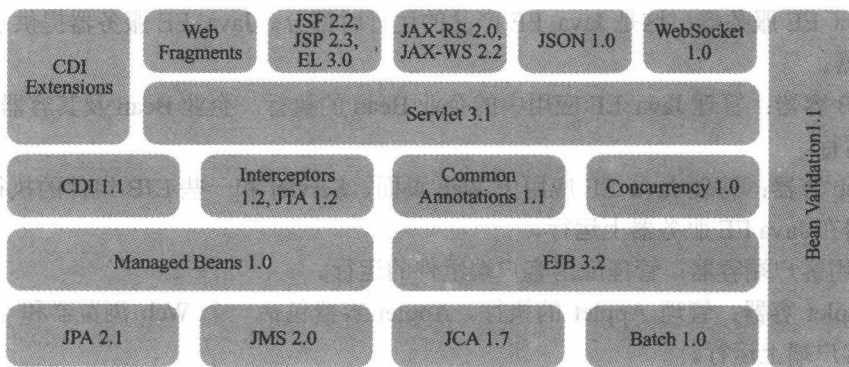


图 1-1 Java EE 7 框架组件

Java EE 利用这些组件给出了“应用程序的模型框架”，该框架把服务实现为多层应用，

从而提供企业级应用开发的可扩展性、可访问性和管理性。一般分为：Domain Object（领域对象）层、DAO（Data Access Object，数据访问对象）层、业务逻辑层、控制层和表现层。

1) Domain Object 层：领域对象组件抽象了系统的对象模型。该层由“领域对象组件”构成，即由系列的 POJO（Plain Old Java Object，即普通的 Java 对象）组成，往往包含了各自所需要的逻辑方法。

2) DAO 层：此层由系列的 DAO 组件组成，这些 DAO 实现了对数据库的查询、更新、删除和创建（CRUD）等原子操作。

3) 业务逻辑层：此层由系列的业务逻辑对象组成，这些业务逻辑对象实现了系统所需要的业务逻辑方法。

4) 控制层：此层由系列控制器组成，这些控制器用于拦截用户请求，并调用业务逻辑组件的业务逻辑方法，处理用户请求，并根据处理结果转发到不同的表现层组件。

5) 表现层：该层主要负责收集用户请求，并显示处理结果。最常用的技术是 JSP 技术和 JSF 技术。

各层之间的关系如图 1-2 所示，各层的 Java EE 组件以松耦合的方式耦合在一起，这种方式方便了应用程序的扩展性，逆着箭头的方向，前面的组件依赖于后面的组件的功能；顺着箭头的方向，前面的组件支持后面的组件实现。

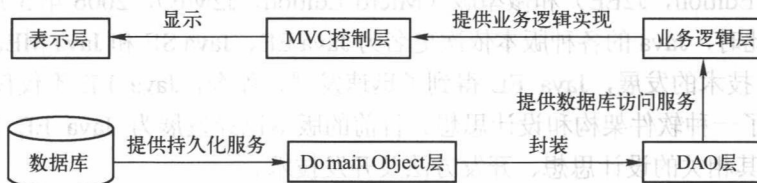


图 1-2 Java EE 应用模型结构框架

1.1.2 Java EE API 及主要技术

Java EE 以“容器”的形式提供服务，不同的容器由不同的组件构成，主要的服务器和容器有：Java EE 服务器、EJB 容器、Web 容器、应用客户端容器和 Applet 容器。

1) Java EE 服务器：这是 Java EE 产品的运行时部分。Java EE 服务器提供了 EJB 容器和 Web 容器。

2) EJB 容器：管理 Java EE 应用中的企业 Bean 的执行。企业 Bean 及其容器在 Java EE 服务器上运行。

3) Web 容器：管理 Java EE 应用中 Web 页面、Servlet 和一些 EJB 组件的执行。Web 组件及其容器在 Java EE 服务器上运行。

4) 应用客户端容器：管理应用客户端组件的运行。

5) Applet 容器：管理 Applet 的执行。Applet 容器包括一个 Web 浏览器和一个 Java 插件，都在客户端上运行。

图 1-3 给出了 Java EE 中各容器之间的关系，以及每个容器中可用的 Java EE 7 的 API。其中图 1-3a 表示容器之间的关系，图 1-3b 显示应用客户端容器中可用的 API；图 1-3c 显示 Web 容器中可用的 API；图 1-3d 显示 EJB 容器中可用的 API。

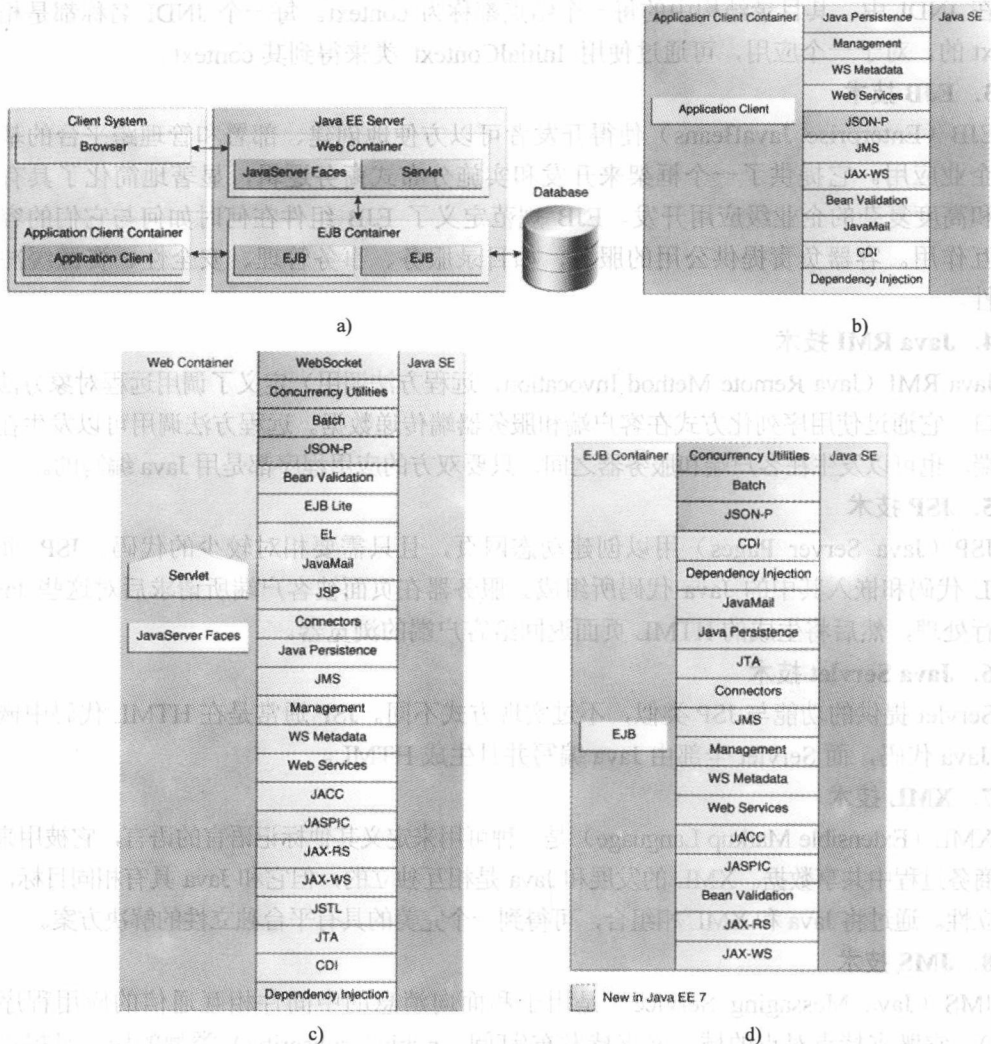


图 1-3 Java EE 中各容器之间的关系及可用的 API

a) 各容器之间的关系 b) 应用客户端容器中可用的 API c) Web 容器中可用的 API d) EJB 容器中可用的 API

从图 1-3 可以看出，Java EE 7 涉及的技术有：JDBC、JNDI、EJB、RMI、JSP、Servlet、XML、JMS、Java IDL、JTS、JTA、JavaMail 和 JAF 等，下面对其中的主要技术给出简单的介绍。

1. JDBC 技术

JDBC (Java Database Connectivity) 提供连接各种关系数据库的统一接口，可以为多种关系数据库提供统一访问，它由一组用 Java 语言编写的类和接口组成。JDBC 为数据库开发人员提供了一个标准的 API。具体应用将在第 2 章给出应用案例。

2. JNDI 技术

JNDI (Java Name and Directory Interface, Java 命名和目录接口) 被用于执行名称和目录服务。它提供了一致的模型来存取和操作企业级的资源，如 DNS、LDAP、本地文件和目录服务。

在 JNDI 中，其目录结构中的每一个结点都称为 context。每一个 JNDI 名称都是相对于 context 的。对于一个应用，可通过使用 InitialContext 类来得到其 context。

3. EJB 技术

EJB (Enterprise JavaBeans) 使得开发者可以方便地创建、部署和管理跨平台的基于组件的企业应用。它提供了一个框架来开发和实施分布式商务逻辑，显著地简化了具有可伸缩性和高度复杂的企业级应用开发。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，如目录服务、事务管理、安全性、资源缓冲池及容错性。

4. Java RMI 技术

Java RMI (Java Remote Method Invocation, 远程方法调用) 定义了调用远程对象方法的标准接口。它通过使用序列化方式在客户端和服务器端传递数据。远程方法调用可以发生在对等的两端，也可以发生在客户端和服务器之间，只要双方的应用程序都是用 Java 编写的。

5. JSP 技术

JSP (Java Server Pages) 用以创建动态网页，且只需要相对较少的代码。JSP 页面由 HTML 代码和嵌入其中的 Java 代码所组成。服务器在页面被客户端所请求后对这些 Java 代码进行处理，然后将生成的 HTML 页面返回给客户端的浏览器。

6. Java Servlet 技术

Servlet 提供的功能与 JSP 类似，不过实现方式不同。JSP 通常是在 HTML 代码中嵌入少量的 Java 代码，而 Servlet 全部由 Java 编写并且生成 HTML。

7. XML 技术

XML (Extensible Markup Language) 是一种可用来定义其他标记语言的语言，它被用来在不同的商务过程中共享数据。XML 的发展和 Java 是相互独立的，但它和 Java 具有相同目标，即平台独立性。通过将 Java 和 XML 相组合，可得到一个完美的具有平台独立性的解决方案。

8. JMS 技术

JMS (Java Messaging Service) 是用于和面向消息的中间件相互通信的应用程序接口 (API)。它既支持点对点的域，又支持发布/订阅 (publish/subscribe) 类型的域，且提供对下列类型的支持：经认可的消息传递、事务型消息的传递、一致性消息和具有持久性的订阅者支持。

9. Java IDL 技术与 CORBA 技术

在 Java IDL 的支持下，开发人员可以将 Java 和 CORBA 集成在一起。它们可创建 Java 对象并使之可在 CORBA ORB 中展开，或可创建 Java 类并作为和其他 ORB 一起展开的 CORBA 对象的客户。后一种方法提供了另外一种途径，通过它 Java 可以被用于将新的应用和 legacy 系统相集成。

10. JTA 技术和 JTS 技术

JTA (Java Transaction Architecture) 定义了一种标准的 API，应用系统由此可存取各种事务监控。

JTS (Java Transaction Service) 是 CORBA OTS 事务监控的基本的实现，JTS 规定了事务管理器的实现方式。该事务管理器是在高层支持 Java Transaction API (JTA) 规范，且在较低层实现 OMG OTS specification 的 Java 映像。JTS 事务管理器为应用服务器、资源管理