



高等学校计算机科学与技术教材

Java EE 编程技术

(第2版)

COMPUTER Science and Technology

□ 郝玉龙 编著

- 原理与技术的完美结合
- 教学与科研的最新成果
- 语言精练，实例丰富
- 可操作性强，实用性突出



清华大学出版社

● 北京交通大学出版社

013071446

TP312JA-43
93-2

高等学校计算机科学与技术教材

Java EE 编程技术

(第2版)

郝玉龙 编著



北航 C1678642

清华大学出版社
北京交通大学出版社

·北京·

TP312JA-43
93-2

内 容 简 介

本书对 Java EE 6 编程技术进行了系统介绍。首先对 Java EE 体系结构进行概述,介绍 Java EE 的定义、设计思想、技术框架等,然后详细指导读者如何基于 NetBeans+MySQL 来搭建 Java EE 开发环境,随后以 Java EE 企业应用的表现层、业务逻辑层和数据持久化层的编程任务为主线,由浅入深地讲解各个应用层次开发相关的组件技术、服务技术、通信技术和架构技术等,重点对 Java EE 6 规范最新功能特性包括 Servlet 3.0、JSF 2.0、EJB 3.1 和 JPA 2.0 等的讲解演示。

本书在第 1 版的基础上对原有内容进行了调整,增加了 Java EE 6 规范中 JSF、JPA、CDI 和 Bean Validation 等关键内容,同时对 Servlet 和 EJB 等部分内容进行了较大补充。

本书适合作为高等学校计算机专业教材,也可作为相关人员的参考书。本书每一章都是一个完整独立的部分,因此教师在授课时可根据授课重点、课时数量进行灵活调整。

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 13501256678 13801310933

图书在版编目(CIP)数据

Java EE 编程技术 / 郝玉龙编著. —2 版. —北京:清华大学出版社;北京交通大学出版社, 2013.6

(高等学校计算机科学与技术教材)

ISBN 978-7-5121-1489-0

I. ①J… II. ①郝… 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(2013)第 121154 号

责任编辑:谭文芳

出版发行:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>
北京交通大学出版社 邮编:100044 电话:010-51686414 <http://www.bjtup.com.cn>

印刷者:北京时代华都印刷有限公司

经 销:全国新华书店

开 本:185×260 印张:23.75 字数:590 千字

版 次:2013 年 6 月第 2 版 2013 年 6 月第 1 次印刷

书 号:ISBN 978-7-5121-1489-0/TP·746

印 数:1~4 000 册 定价:43.00 元

本书如有质量问题,请向北京交通大学出版社质监局反映。对您的意见和批评,我们表示欢迎和感谢。
投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@bjtu.edu.cn。

第2版前言

为什么写作本书

随着社会信息化程度不断提高,越来越多的软件开发人员需要开发企业级应用程序。目前企业应用开发主要有两大技术体系:基于 Java EE 的应用开发和基于 .NET 的应用开发。Java EE 技术以其开放性、灵活性、安全性和技术成熟度,赢得了广大编程爱好者的青睐,熟练掌握 Java EE 编程技术已经成为软件开发人员的重要技能。

2005 年 3 月出版的《J2EE 编程技术》一书,坚持理论与实践相结合的原则,既注重 Java EE 基本理论的讲解,又注重 Java EE 编程技术在开发应用软件时的实践运用。因此,该书在实际教学实践中取得了很好的效果,赢得了广大读者的一致好评。2008 年 6 月,《J2EE 编程技术》第一次修订,由于自 2005 年 J2EE 5.0 版本推出以后, Sun 正式将 J2EE 的官方名称改为“Java EE”,因此 2008 年修订后的书名改为《Java EE 编程技术》。

Java EE 编程技术是一门飞速发展的新技术,特别是 Java EE 6 规范的提出,使得 Java EE 编程变得更加高效、轻松。为了更好地反映 Java EE 编程技术的新发展,本书在《Java EE 编程技术》一书的基础上进行再次改版。

内容调整

在与出版社编辑和多位任课老师反复交流后,本着“确保体系完整、面向实际应用,反映技术趋势”的指导原则,对本书的内容进行了如下调整。

Servlet 编程部分增加了针对 Servlet 3.0 的新功能特性,包括异步处理方法、上传附件支持等。

JSP 和 JavaBean 的内容基本保持不变。

EJB 编程部分增加了针对 EJB 3.1 的新功能特性,包括单例 Bean、Time 服务、拦截器和异步方法支持等。

JDBC 编程部分增加了 RowSet 部分的相关内容。

JSF 和 JPA 作为 Java EE 6 规范的重要组成部分进行了重点讲解。

另外,由于篇幅的限制,去掉了 JSP 标记库和 JavaMail 编程两部分内容。

在综合练习部分针对 Java EE 6 规范的新内容,采用 JSF+EJB+JPA 架构进行了重新编写。

本书的特色

本书最大的特色在于坚持理论与实践相结合的原则,既注重 Java EE 基本原理的讲解,又注重对 Java EE 编程技能的示范,使读者既能够透彻理解 Java EE 基本的原理和概念,又能

够切实提高 Java EE 编程能力。在 Java EE 基本原理讲解方面,结合作者自身理解和体会,以通俗、简练的语言对 Java EE 核心概念和原理进行重点讲解,尽量避免在一些烦琐的技术细节上过多纠缠,不求面面俱到,力争使读者能够在较短的时间里掌握在实际应用开发中必需的基本概念和技术,并对 Java EE 的体系框架有个全面的整体认识。书中所有示例都是作者结合多年教学实践和实际工程项目经验严格挑选的,力求简洁明了,切中要害,使读者能够快速理解并运用到实践中去。

本书的另外一大特色是系统完整、结构合理。Java EE 是一个包含众多开发技术的标准规范,涵盖了企业应用开发的各个层面。本书首先对 Java EE 编程技术进行概述,然后选取 Java EE 编程中最核心的技术进行深入讲解,力求读者在学习后能够对整个 Java EE 技术体系和编程思想获得全面清晰的了解,最后通过一个综合示例来对之前所学的内容进行总结归纳和升华提高。

开发环境的选择

为方便 Java EE 编程技能的示范,本书采用目前流行的免费 Java EE 开发环境 NetBeans IDE 和 MySQL。NetBeans 是目前唯一一个集成了完全兼容 Java EE 6 规范的应用服务器的集成开发环境,减少了在开发环境搭建方便难度,且 NetBeans 对硬件配置要求不高,适应学习培训的需求。

适用读者

本书适合已经掌握了 Java 语言,希望学习 Java EE 开发的读者。由于 Java EE 6 规范中吸收了当今流行框架的设计思想和理念。在学习完本书的内容后再深入学习 Struts2、Spring 和 Hibernate 等流行架构技术将会有更好的效果。

致谢

作为高等院校 Java EE 编程技术的教材,本书受到了国内众多高校教师和学生的关注,许多老师和同学对《Java EE 编程技术》提出了很好的建议和修改意见,在此表示感谢。

在本书的编写过程中,得到众多老师的指导和帮助。感谢解放军理工大学的程宝义教授、张宏军教授、张红雷高工、吴耀平高工,他们为本书提供了良好的技术支持。感谢周旋、尹鹏飞、周铭、张志杰、姜波、雷霆、宋祥斌、尹建平,他们参与了本书的部分编辑修改工作,并对本书的内容组织提供了建设性的意见。感谢孙阳、胡磊、李冰,他们测试了本书的全部代码。感谢本书的编辑谭文芳老师,她对本书再版提供了宝贵的建议。特别感谢我的妻子,在我写作的过程中给我无微不至的关怀。

由于作者水平有限,加之编写时间仓促,书中难免出现错误和不足。对于书中的任何问题,请发 E-mail 至邮箱:haoyulongsd@163.com。

郝玉龙

2013 年 6 月于北京

目 录

第 1 章	Java EE 概述	1
1.1	Java EE 产生的背景	1
1.1.1	企业级应用程序特点	1
1.1.2	企业级应用程序体系结构	2
1.2	什么是 Java EE	3
1.3	Java EE 编程思想: 组件-容器	5
1.4	Java EE 技术框架	6
1.4.1	组件技术	6
1.4.2	服务技术	7
1.4.3	通信技术	9
1.4.4	架构技术	10
1.5	Java EE 体系架构的优点	11
	小结	11
	习题 1	11
第 2 章	搭建开发环境	12
2.1	概述	12
2.2	安装 JDK	12
2.3	安装 NetBeans IDE	14
2.4	开发环境测试	17
	小结	19
	习题 2	19
第 3 章	Servlet	20
3.1	Servlet 基础	20
3.1.1	Servlet 定义	20
3.1.2	Servlet 工作流程	21
3.1.3	Servlet 编程接口	23
3.2	第一个 Servlet	24
3.3	处理请求	32
3.3.1	请求参数	32
3.3.2	Header	38
3.3.3	上传文件	40
3.3.4	异步处理	42

3.4	生成响应	45
3.4.1	编码类型	45
3.4.2	流操作	46
3.4.3	重定向	48
3.5	Servlet 配置	49
3.5.1	初始化参数	49
3.5.2	URL 模式	51
3.5.3	默认 Servlet	51
3.6	会话管理	52
3.6.1	Cookie	52
3.6.2	URL 重写	55
3.6.3	HttpSession	58
3.7	Servlet 上下文	60
3.8	Servlet 间协作	64
3.9	Filter	68
3.10	Listener	74
	小结	83
	习题 3	84
第 4 章	JSP	85
4.1	概述	85
4.2	第一个 JSP	86
4.3	脚本	87
4.3.1	输出表达式	87
4.3.2	注释	88
4.3.3	声明变量、方法和类	89
4.4	指令	93
4.4.1	page 指令	93
4.4.2	include 指令	96
4.5	动作组件	98
4.6	内置对象	105
4.6.1	request 对象	105
4.6.2	response 对象	113
4.6.3	session 对象	115
4.6.4	application 对象	118
4.6.5	out 对象	120
4.6.6	exception 对象	121
4.6.7	内置对象的作用范围	122
4.7	表达式语言	123
4.7.1	基本语法	123

4.7.2	隐式对象	125
4.7.3	存取器	125
	小结	127
	习题 4	127
第 5 章	JavaBean	128
5.1	JavaBean 原理与应用	128
5.1.1	什么是 JavaBean	128
5.1.2	JavaBean 的简单例子	128
5.2	JavaBean 和 JSP 的结合	132
5.2.1	JSP 中 JavaBean 相关标记	132
5.2.2	JavaBean 的生命周期范围	137
5.3	完整示例	138
	小结	140
	习题 5	140
第 6 章	JSF	141
6.1	什么是 JSF	141
6.1.1	什么是框架	142
6.1.2	为什么需要框架	142
6.1.3	什么是 JSF	143
6.1.4	为什么学习 JSF	144
6.2	第一个 JSF 应用	144
6.2.1	创建 JSF 项目	144
6.2.2	模型组件	146
6.2.3	视图组件	147
6.2.4	控制组件	148
6.2.5	运行演示	149
6.2.6	总结思考	149
6.3	Managed Bean	150
6.3.1	定义 Managed Bean	150
6.3.2	生命周期	152
6.3.3	CDI Bean	156
6.3.4	Bean 之间的依赖	158
6.4	视图	160
6.4.1	组件树	160
6.4.2	标记	161
6.4.3	EL 支持	165
6.4.4	资源管理	166
6.5	页面模板	168
6.6	JSF 请求处理过程	172

6.6.1	常规流程	172
6.6.2	示例分析	173
6.6.3	特殊流程	177
6.6.4	异常处理	178
6.6.5	总结思考	181
6.7	导航控制	182
6.7.1	导航组件	182
6.7.2	Action 方法	182
6.7.3	导航约定	184
6.7.4	导航规则	184
6.7.5	重定向	187
6.7.6	Get 请求	189
6.7.7	JSF 框架外导航	192
6.7.8	导航中的参数传递	193
6.8	类型转换	196
6.8.1	标准转换器	196
6.8.2	自定义转换器	197
6.9	输入校验	200
6.9.1	标准校验器	201
6.9.2	自定义校验器	202
6.9.3	Bean 方法校验	203
6.9.4	Bean 验证框架	203
6.10	事件处理	204
6.10.1	Value Change 事件	205
6.10.2	Action 事件	208
6.10.3	Phase 事件	210
6.10.4	System 事件	212
6.11	国际化支持	214
6.11.1	准备资源包	215
6.11.2	配置资源包	216
6.11.3	在 JSF 视图中使用资源	217
6.11.4	设置应用程序本地属性	218
6.11.5	异常消息本地化	219
6.12	Ajax	220
	小结	223
	习题 6	223
第 7 章	JDBC	224
7.1	搭建 JDBC 开发环境	224
7.1.1	安装数据库系统	224

7.1.2	安装驱动程序	226
7.2	连接数据库	226
7.3	执行 SQL 语句	229
7.3.1	Statement	230
7.3.2	PreparedStatement	233
7.3.3	CallStatement	237
7.4	ResultSet	239
7.4.1	光标	239
7.4.2	BLOB 字段处理	240
7.5	RowSet	241
7.6	连接池和数据源	245
7.6.1	创建 MySQL 数据库的连接池	247
7.6.2	创建数据源	249
7.6.3	基于数据源访问数据库	250
小结		252
习题 7		252
第 8 章 JPA		253
8.1	概述	253
8.2	第一个 JPA 应用	254
8.2.1	持久化单元	254
8.2.2	创建 Entity	255
8.2.3	利用 EntityManager 操作 Entity	257
8.2.4	运行演示	259
8.3	ORM	259
8.3.1	Entity	260
8.3.2	主键	260
8.3.3	复合主键	261
8.3.4	属性	265
8.3.5	关联映射	267
8.3.6	加载方式	272
8.3.7	顺序	273
8.3.8	继承映射	273
8.4	Entity 管理	274
8.4.1	获取 EntityManager	274
8.4.2	持久化上下文	275
8.4.3	Entity 操作	275
8.4.4	级联操作	283
8.5	JPQL	284
8.5.1	动态查询	285

8.5.2	参数设置	286
8.5.3	命名查询	287
8.5.4	属性查询	288
8.5.5	使用构造器	288
8.6	本地查询	289
8.7	基于 Criteria API 的安全查询	290
8.8	生命周期回调方法	292
8.9	校验框架	294
	小结	295
	习题 8	295
第 9 章	EJB	296
9.1	EJB 基础	296
9.1.1	为什么需要 EJB	296
9.1.2	EJB 容器	297
9.1.3	EJB 组件	299
9.1.4	EJB 接口	300
9.1.5	EJB 分类	300
9.1.6	部署 EJB	300
9.1.7	EJB 优点	301
9.2	无状态会话 Bean	301
9.2.1	无状态会话 Bean	302
9.2.2	开发一个无状态会话 EJB	303
9.2.3	利用 Servlet 测试无状态会话 EJB	309
9.2.4	利用远程客户端测试无状态会话 Bean	312
9.3	有状态会话 Bean	313
9.3.1	基本原理	313
9.3.2	实现有状态会话 Bean	315
9.4	单例 Bean	319
9.4.1	基本原理	319
9.4.2	利用 JSF 访问单例 Bean	320
9.4.3	并发控制	322
9.4.4	依赖管理	323
9.5	消息驱动 Bean	324
9.5.1	基本原理	324
9.5.2	实现消息驱动 Bean	325
9.6	Time 服务	330
9.7	拦截器	333
9.8	异步方法	335
	小结	341

习题 9.....	341
第 10 章 Web 服务	342
10.1 概述.....	342
10.2 什么是 Web 服务.....	343
10.3 Web 服务技术体系.....	343
10.4 Web 服务工作模型.....	344
10.5 Java EE 平台下的 Web 服务实现.....	345
10.6 开发 Web 服务实例.....	346
10.6.1 创建 Web 服务.....	346
10.6.2 部署 Web 服务.....	348
10.6.3 测试 Web 服务.....	348
10.6.4 在 Web 应用程序中调用 Web 服务.....	350
10.7 Web 服务的优缺点.....	354
小结.....	355
习题 10.....	355
第 11 章 综合练习	356
11.1 基础知识.....	356
11.1.1 概述.....	356
11.1.2 架构类型.....	356
11.2 功能需求.....	358
11.3 数据库设计.....	358
11.4 系统整体架构.....	359
11.5 系统实现.....	359
11.5.1 表示逻辑层.....	359
11.5.2 业务逻辑层.....	363
11.5.3 数据表示层.....	364
11.6 运行界面.....	365
小结.....	366

第 1 章 Java EE 概述

本章要点:

- ☑ Java EE 定义
- ☑ Java EE 编程思想
- ☑ Java EE 技术框架
- ☑ Java EE 优点

本章首先讲解 Java EE 的基本概念, 随后对 Java EE 的编程思想、技术框架等内容进行深入分析, 最后对 Java EE 编程技术的优点进行评析。

1.1 Java EE 产生的背景

随着社会信息化程度不断提高, 越来越多的软件开发人员需要开发企业级的应用程序。为了满足开发多层体系结构的企业级应用的需求, Java 的创始人 Sun 公司在早期的 J2SE (Java 2 Platform Standard Edition) 基础上, 针对企业级应用的各种需求, 提出了 J2EE (Java 2 Platform Enterprise Edition)。

说明: 自 2005 年 J2EE 5.0 版本推出以后, Sun 正式将 J2EE 的官方名称改为“Java EE”。因此在本书以后的描述中, 统一使用“Java EE”这一术语。

1.1.1 企业级应用程序特点

所谓的企业级应用程序, 并不是特指为企业开发的应用软件, 而是泛指那些为大型组织部门创建的应用程序。与常见的应用程序相比较, 企业级应用程序一般具有以下特点。

(1) 分布式。企业应用程序通常不是运行在某个单独的 PC 上, 而是通过局域网运行在一个组织内部, 或通过 Internet 连接分布在世界各地的部门或用户。

(2) 高速反应性。社会信息瞬息万变, 企业组织必须不断地改变业务规则来适应社会信息的高速变化, 相应地, 对应用程序也不断提出新的需求。企业应用程序必须具备能力来及时适应需求的变化, 同时又尽可能地减少资金的投入。

(3) 安全性。实现应用系统的正常操作和运转, 对于企业的成功来说至关重要。但仅仅做到这一点还不够, 还必须保证系统运行的安全可靠。

(4) 可扩展性。在网络环境内, 一个应用的潜在用户可能有成百上千, 在这种情况下, 企业应用除了要考虑能够更加有效地利用企业不断增长的信息资源外, 还要充分考虑用户群体的膨胀给应用带来的性能上的扩展需求。

(5) 集成化。信息是企业资产的重要部分, 但目前企业多数信息作为数据存放在老的

或已经过时的应用系统中。为了最大限度地利用信息资源，要求新的应用必须与目前存在的遗留应用系统相互集成。由于新老系统间采用的技术、运行的平台都可能不同，因此这种集成要求对于应用开发来说是个很大的挑战。

1.1.2 企业级应用程序体系结构

应用程序体系结构是指应用程序内部各组件间的组织方式。企业级应用程序的体系结构的设计经历了从两层结构到三层结构再到多层结构的演变过程。

1. 两层体系结构应用程序

如图 1-1 所示，两层体系结构应用程序共分为客户层（Client）和服务器层（Server），因此又称为 C/S 模式。在两层体系结构中，客户层的客户端程序负责实现人机交互、应用逻辑、数据访问等职能；服务器层由数据库服务器来实现，唯一职能是提供数据库服务。这种体系结构应用程序有以下缺点。

（1）安全性低。客户端程序与数据库服务器直接连接，非法用户容易通过客户端程序侵入数据库，造成数据损失。

（2）部署困难。集中在客户端的应用逻辑导致客户端程序肥大，而且随着业务规则的不断变化，需要不断更新客户端程序，大大增加了程序部署工作量。

（3）耗费系统资源。每个客户端程序都要直接连到数据库服务器，使服务器为每个客户端建立连接而消耗大量宝贵的服务器资源，导致系统性能下降。

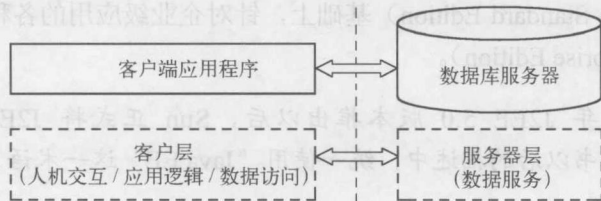


图 1-1 两层体系结构应用程序

2. 三层体系结构应用程序

为解决两层体系结构应用程序带来的问题，软件开发领域又提出三层体系结构应用程序，在二层体系结构应用程序的客户层与服务器层之间又添加了一个第三层——应用服务器层。这样，应用程序共分为客户层、应用服务器层和数据服务器层三个层次，如图 1-2 所示。与两层体系结构的应用相比，三层体系结构应用程序的客户层功能大大减弱，只用来实现人机交互，原来由客户端实现的应用逻辑、数据访问职能都迁移到应用服务器层上来实现，因此客户层通常被称做“瘦客户层”。数据服务器层仍旧仅提供数据信息服务。由于客户层应用程序通常由一个通用的浏览器（Browser）程序实现，因此这种体系结构又被称做 B/S 模式或“瘦客户机”模式。应用服务器层是位于客户层与数据服务器层中间的一层，因此应用服务器被称做“中间件服务器”或“中间件”，应用服务器层又被称做“中间件服务器层”。

相对于两层体系结构的应用程序，三层体系结构的应用程序具有以下优点。

（1）安全性高。中间件服务器层隔离了客户端程序对数据服务器的直接访问，保护了

数据信息的安全。

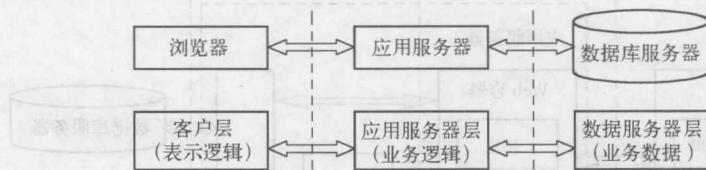


图 1-2 三层结构的应用程序

(2) 易维护。由于业务逻辑在中间件服务器上，当业务规则变化后，客户端程序基本不做改动，只需要升级应用服务器层的程序即可。

(3) 快速响应。通过中间件服务器层的负载均衡以及缓存数据能力，可以大大提高对客户端的响应速度。

(4) 系统扩展灵活。基于三层分布体系的应用系统，可以通过在应用服务器部署新的程序组件来扩展系统规模；当系统性能降低时，可以在中间件服务器层部署更多的应用服务器来提升系统性能，缩短客户端的响应。

3. 多层体系结构应用程序

可以将中间件服务器层按照程序应用逻辑进一步划分为若干个子层，这样就形成了多层体系结构的应用程序。关于多层体系结构应用程序，类似于三层体系结构，这里不再赘述。在有些文献中也将三层以及三层以上体系结构应用程序统称为多层体系结构应用程序。

为了满足开发多层体系结构的企业级应用的需求，Sun 公司在早期的 J2SE 基础上，针对企业级应用的各种需求提出了 Java EE。

1.2 什么是 Java EE

在深入学习 Java EE 之前，首先要明确什么是 Java EE。

1. Java EE 是一个标准中间件体系结构

不要被名称“Java Platform Enterprise Edition”误导，与 Java 不同，Java EE 是一种体系结构，而不是一门编程语言。Java 是一门编程语言，可以用来编写各种应用程序。Java EE 是一个标准中间件体系结构，旨在简化和规范分布式多层企业应用系统的开发和部署。

Java EE 出现之前，分布式多层企业应用系统的开发和部署没有一个被普遍认可的行业标准，几家主要的中间件开发商的产品各自为政，彼此之间缺乏兼容性，可移植性差，难以实现互操作。Java EE 的出现，规范了分布式多层体系的应用开发。Java EE 将企业应用程序划分为多个不同的层，并在每一个层上定义对应的组件来实现它。典型的 Java EE 结构的应用程序包括四层：客户层、表示逻辑层（Web 层）、业务逻辑层和企业信息系统层，如图 1-3 所示。

客户层可以是网络浏览器或者是桌面应用程序。

表示逻辑层（Web 层）、业务逻辑层都位于应用服务器上，它们都是由一些 Java EE 标准组件 JSP（Java Server Page）、JSF（Java Server Face）、Servlet、EJB（Enterprise JavaBeans）和 Entity 等来实现，这些组件运行在实现了 Java EE 标准的应用服务器上，以实现特定的表现逻辑和业务逻辑。

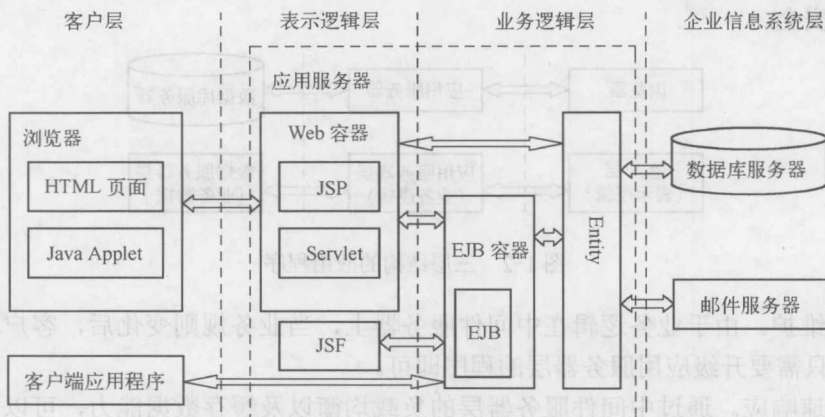


图 1-3 Java EE 多层体系结构

企业信息系统层主要用于企业信息的存储管理，主要包括数据库系统、电子邮件系统、目录服务等。Java EE 应用程序组件经常需要访问企业信息系统层来获取所需的数据信息。

Java EE 体系架构的实施可显著地提高企业应用系统的可移植性、安全性、可伸缩性、负载平衡和可重用性。

2. Java EE 是企业分布式应用开发标准

Java EE 本身是一个为企业分布式应用开发提供的标准，Java 技术标准组织领导着 Java EE 规范和标准的制定。Java EE 规范描述了 Java EE 结构，可以从网址 <http://download.oracle.com/javaee/6/api/> 下载最新的 Java EE 规范。Java EE 作为一个企业分布式应用开发标准，主要体现在以下两个方面：

(1) Java EE 规范了分布式组件开发的标准。Java EE 包含的分布式组件有 JSP、Servlet、EJB 等。Java EE 标准规定了这些分布式组件应该实现哪些接口，应该提供哪些方法。

(2) Java EE 规范了分布式组件运行环境——容器提供的服务标准。分布式组件的运行环境称为容器，容器通过提供标准的服务来支持分布式组件的运行。不同的分布式组件由不同的容器来支撑运行，如 JSP 和 Servlet 运行在 Web 容器中，EJB 组件运行在 EJB 容器中。Java EE 规范中，容器实现的标准服务有安全、事务管理、上下文和依赖注入、校验和远程连接等。

截止到 2012 年 9 月，Java EE 的最新版本为 6。Java EE 6 规范已被众多中间件开发厂商接受并实现。实现 Java EE 6 完整规范的应用服务器有 Oracle 的 Weblogic 12c、IBM 的 Websphere V8，同时还有其他一些免费软件，如 JBoss AS 7.1 和 GlassFish 3.1 等。

说明：随着 Java EE 版本的升级，它所包含的技术规范越来越多。为了降低 Java EE 规范支持的难度，Java EE 中提出了 Profile 的概念。Profile 是针对特定应用领域的技术规范子集，它剪切掉一些很少使用的技术，使得 Java EE 变得更加简洁，也便于开发商实现。目前 Java EE 规范中支持的唯一一个 Profile 是 Web Profile。例如 Apache Tomcat 就是仅仅实现了 Java EE 6 Web Profile 的应用服务器。

尽管不同的厂家有不同的产品实现，但它们都遵循同一个 Java EE 规范。因此遵循

Java EE 标准的分布式组件，可以自由部署在这些由不同厂商生产的、但相互兼容的 Java EE 容器环境内。分布式系统的开发变得简单，部署的速度也大大加快。

1.3 Java EE 编程思想：组件 - 容器

Java EE 为满足开发多层体系结构的企业级应用的需求，提出“组件-容器”的编程思想。Java EE 应用的基本软件单元是 Java EE 组件。所有的 Java EE 组件都运行在特定的运行环境之中。组件的运行环境被称为容器。Java EE 组件分为 Web 组件和 EJB 组件，相应地，Java EE 容器也分为 Web 容器和 EJB 容器。

容器为组件提供必需的底层基础功能，容器提供的底层基础功能被称为服务。组件通过调用容器提供的标准服务来与外界交互。为满足企业级应用灵活部署，组件与容器之间必须既松散耦合，又能够强有力地交互。为实现这一点，组件与容器都要遵循一个标准规范。这个标准规范就是 Java EE。

Java EE 容器由专门的厂商来实现，容器必须实现的基本接口和功能由 Java EE 规范定义，但具体如何实现完全由容器厂商自己决定。常见的 Java EE 服务器中都包含了 Web 容器或 EJB 容器的实现。组件一般由程序员根据特定的业务需求编程实现。

所有的 Java EE 组件都是在容器的 Java 虚拟机中进行初始化的，组件通过调用容器提供的标准服务来与外界交互。容器提供的标准服务有命名服务、数据库连接、持久化、Java 消息服务、事务支持、安全服务等。因此在分布式组件的开发过程中，完全可以不用考虑复杂多变的分布式计算环境，而专注于业务逻辑的实现，这样可大大提高了组件开发的效率，降低开发企业级应用程序的难度。

那么组件与容器之间是如何实现交互的呢？即容器如何知道要为组件提供何种服务，组件又是如何来获取容器提供的服务呢？Java EE 采用部署描述文件来解决这一难题。每个发布到服务器上的应用除了要包含自身实现的代码文件外，还要包括一个 XML 文件，该文件被称为部署描述文件。部署描述文件中详细地描述了应用中的组件所要调用的容器服务的名称、参数，等等。部署描述文件就像组件与容器间达成的一个“契约”，容器根据部署描述文件的内容为组件提供服务，组件根据部署文件中的内容来调用容器提供的服务。

从上面的介绍中可以感觉，部署描述文件的配置是 Java EE 开发中的一项重要而又烦琐的工作。值得庆幸的是，自 Java EE 5 规范推出以来，Java EE 支持在组件的实现代码中引入注解来取代配置复杂的部署描述文件。所谓的注解是 JDK 5 版本后支持的一种功能机制，它支持在 Java 组件的源代码中嵌入元数据信息，在部署或运行时应用服务器将根据这些元数据对组件进行相应的部署配置。关于注解，后面的章节中还会详细论述。容器在组件部署时通过提取注解信息来决定如何为组件提供服务。注解的出现大大简化了 Java EE 应用程序的开发和部署，是 Java EE 规范的一项重大进步。

更值得一提的是，在最新的 Java EE 6 规范中，还引入了一种“惯例优于配置”或者称为“仅异常才配置”的思想。通俗一点讲，就是对于 Java EE 组件的一些属性和行为，容器将按照一些约定俗成的惯例来自动进行配置，此时开发人员甚至连注解都可以省略。只有当组件的属性和行为不同于惯例时，才需要进行配置。这种编程方式大大减少了程序人员的工作量，也是需要开发人员逐渐熟悉和适应的一种编程技巧。