

高等学校计算机科学与技术教材

- ① 原理与技术的完美结合
- ② 教学与科研的最新成果
- ③ 语言精炼，实例丰富
- ④ 可操作性强，实用性突出

Java EE

编程技术

□ 郝玉龙 姜犇 编著

清华大学出版社



北京交通大学出版社





高等学校计算机科学与技术教材

Java EE 编程技术

郝玉龙 姜 桦 编著

清华大学出版社

北京交通大学出版社

·北京·

内 容 简 介

本书对 Java EE 编程技术进行了系统的介绍。本书首先对 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 三大组件技术 (Servlet, JSP, EJB) 为主线, 中间穿插 Java EE 服务技术、通信技术对 Java EE 编程技术进行了系统介绍; 最后以一个完整信息系统的开发为示例, 从系统的需求分析开始, 到规划、框架设计、编码, 再到部署发布, 一步步引导读者完成一个完整系统的开发, 向读者展示灵活运用 Java EE 技术构建完整信息系统的基本技巧。

本书是在《J2EE 编程技术》一书的基础上补充修订而成的。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”, 因此修订后的书名改为《Java EE 编程技术》。

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, 姜铎编著. —北京: 清华大学出版社; 北京交通大学出版社, 2008.6

(高等学校计算机科学与技术教材)

ISBN 978-7-81123-241-7

I. J… II. ①郝… ②姜… 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 (2008) 第 027228 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://press.bjtu.edu.cn>

印刷者: 北京交大印刷厂

经 销: 全国新华书店

开 本: 185×260 印张: 22 字数: 563 千字

版 次: 2008 年 6 月第 1 版 2008 年 6 月第 1 次印刷

书 号: ISBN 978-7-81123-241-7/TP·406

印 数: 1~5000 册 定价: 34.00 元

本书如有质量问题, 请向北京交通大学出版社质监组反映。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。

前 言

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 Web 应用程序。目前网络编程主要有两大技术体系：基于 Java EE 的网络开发和基于 .NET 的网络开发。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，并且在目前企业级信息系统开发领域也占领了很大的市场，取得了许多成功的案例。

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，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。

但是由于 Java EE 编程技术是一门飞速发展的新技术，从《J2EE 编程技术》一书出版至今，Java EE 企业应用开发领域发生了重大变化。特别是 Java EE 5.0 规范的推出，更是为 Java 开发技术注入了新的动力。由于自 2005 年 J2EE 5.0 版本推出以后，Sun 公司正式将 J2EE 的官方名称改为“Java EE”，因此修订后的书名改为《Java EE 编程技术》。

为了更好地反映 Java EE 编程技术的发展，本书在《J2EE 编程技术》一书的基础上进行以下修订和补充。一是开发环境的调整，由于 Eclipse 结合免费插件的方式搭建开发环境相对复杂，而且很容易产生版本兼容问题，因此，在本书中改用免费的集成开发环境 Netbeans IDE；二是针对新的 Java EE 5.0 规范，对本书原有的章节内容进行修订补充。

本书主要包括以下内容。

第 1 章 Java EE 概述：介绍 Java EE 产生的背景、Java EE 定义、Java EE 编程思想、技术框架和 Java EE 体系结构的优点等。

第 2 章搭建 Java EE 开发环境：详细指导读者如何基于 Netbeans IDE 和 JDK 来搭建 Java EE 开发环境。

第 3 章 Servlet 编程：讲解 Servlet 的基本概念、工作原理，以及处理客户端输入、获取配置信息、发送非 HTML 文档等 Servlet 编程基本技能和会话管理、Servlet 间协同、Servlet 上下文、过滤器、监听器等高级编程技巧。

第 4 章 JSP 编程：依据新的 JSP 2.0 规范，介绍 JSP 基本语法包括脚本、指令和动作组件、内置组件和表达式语言等。

第 5 章在 JSP 页面中使用 JavaBean：介绍如何将 JavaBean 与 JSP 结合进行 Web 应用开发。

第 6 章 JSP 自定义标记：介绍自定义标记工作原理及如何开发自定义标记库。

第 7 章 JSP 标准标记库：系统介绍 JSP 标准标记库的语法和使用。

第 8 章 数据库编程：介绍在 Java EE 中如何对关系数据库进行操作访问。

第 9 章 Java Mail 编程：介绍如何利用 Java Mail 进行电子邮件的收发及对邮件附件的处理等。

第 10 章 EJB 编程：依照新的 EJB 3.0 规范对 EJB 工作原理及如何开发 EJB 组件进行系

统介绍。

第 11 章 Web 服务编程：系统介绍 Web 服务的基本概念和编程方法。

第 12 章 综合练习：以一个完整信息系统的开发为示例，展示灵活运用 Java EE 技术构建企业级信息系统的基本技巧。

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。在 Java EE 编程技能的示范方面，采用目前最流行的免费的 Java EE 开发环境 NetBeans IDE 和 MySQL，从开发环境的搭建配置到程序的发布运行，一步步引导读者掌握 Java EE 编程开发技能。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。

为方便广大读者使用本书学习 Java EE 编程，可以去北京交通大学出版社的网站 <http://press.njtu.edu.cn/>或网址 <http://www.91files.com/?7WOIWU19HMYLJDGF37DR> 去下载本书相关的源代码，以及关于如何使用这些源代码的详细指导和说明。

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、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。感谢本书的编辑，北京交通大学的谭文芳老师，没有她的辛勤劳动，本书不可能出版。特别感谢我的父母和妻子，在我写作的过程中给我无微不至的关怀。

作为高等院校 Java EE 编程技术的教材，本书的前身《J2EE 编程技术》受到了国内众多高校教师和学生的关注，许多老师和同学对于《J2EE 编程技术》提出了很好的建议和修改意见，在此对他们一并表示感谢。

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

2008 年 3 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	5
1.4.1 组件技术	5
1.4.2 服务技术	7
1.4.3 通信技术	8
1.5 Java EE 体系架构的优点	9
小结	10
习题 1	10
第 2 章 搭建 Java EE 开发环境	11
2.1 概述	11
2.2 安装 JDK	11
2.3 安装 Netbeans IDE	13
2.4 开发环境测试	15
小结	17
习题 2	17
第 3 章 Servlet 编程	18
3.1 Servlet 基础	18
3.1.1 Servlet 定义	18
3.1.2 Servlet 工作原理	19
3.1.3 Servlet 编程接口	20
3.2 第一个 Servlet	21
3.3 处理客户端输入	29
3.4 发送非 HTML 文档	36
3.5 获取 Servlet 配置参数	39
3.6 会话管理	42
3.6.1 使用 Cookie 进行会话跟踪	43
3.6.2 使用 URL 重写跟踪会话	47
3.6.3 使用 HttpSession 对象跟踪会话	50

3.6.4	使用会话实现购物车程序	53
3.7	Servlet 上下文	58
3.8	Servlet 间协作	62
3.9	Filter	67
3.10	Listener	74
	小结	83
	习题 3	83
第 4 章	JSP 编程	84
4.1	概述	84
4.2	第一个 JSP	84
4.3	JSP 脚本元素	86
4.3.1	输出表达式	86
4.3.2	注释	87
4.3.3	声明变量、方法和类	88
4.4	指令	92
4.4.1	page 指令	92
4.4.2	include 指令	95
4.5	JSP 动作组件	96
4.6	JSP 内置对象	103
4.6.1	request 对象	104
4.6.2	response 对象	112
4.6.3	session 对象	114
4.6.4	application 对象	117
4.6.5	out 对象	119
4.6.6	exception 对象	119
4.6.7	内置对象的作用范围	120
4.7	表达式语言	122
4.7.1	基本语法	122
4.7.2	隐式对象	123
4.7.3	存取器	124
	小结	126
	习题 4	126
第 5 章	在 JSP 页面中使用 JavaBean	127
5.1	JavaBean 原理与应用	127
5.1.1	什么是 JavaBean	127
5.1.2	JavaBean 的简单例子	127
5.2	JavaBean 和 JSP 的结合	131
5.2.1	JSP 中 JavaBean 相关标记	131
5.2.2	关于 JavaBean 的 Scope 属性的说明	135

5.3 JSP 与 JavaBean 结合的简单例子	136
小结	138
习题 5	138
第 6 章 JSP 自定义标记	139
6.1 基础知识	139
6.1.1 什么是自定义标记	139
6.1.2 自定义标记的表示和分类	140
6.1.3 自定义标记工作原理	140
6.1.4 自定义标记库	141
6.1.5 自定义标记库组成	142
6.2 开发第一个自定义标记	143
6.3 建立带有属性的标记	147
6.4 建立处理体内容的标记	152
6.5 引入脚本变量的标记	156
6.6 开发嵌套标记	161
6.7 标记库的发布	165
小结	170
习题 6	170
第 7 章 JSP 标准标记库	171
7.1 JSTL 基础	171
7.1.1 概述	171
7.1.2 JSTL 组成	172
7.1.3 使用 JSTL	172
7.2 core 标记库	173
7.2.1 通用标记	174
7.2.2 流程控制标记	178
7.2.3 URL 标记	183
7.3 SQL 标记库	185
7.3.1 获取数据源	185
7.3.2 查询	185
7.3.3 数据库更新	189
7.3.4 事务处理	189
7.4 XML 标记库	190
7.4.1 XML 核心标记	190
7.4.2 转换标记	192
7.4.3 XML 流控制标记	194
7.5 I18N 标记库	197
7.5.1 本地化上下文标记	197
7.5.2 日期标记	200

7.5.3	数字标记	203
7.5.4	消息标记	205
7.6	fn 标记库	209
	小结	210
	习题 7	211
第 8 章	数据库编程	212
8.1	搭建 JDBC 开发环境	212
8.1.1	安装数据库系统	212
8.1.2	安装驱动程序	214
8.2	创建数据库连接	214
8.2.1	注册驱动程序	215
8.2.2	JDBC URL	215
8.2.3	建立连接	215
8.3	创建和执行简单 SQL 语句	217
8.4	执行带参数的 SQL 语句	221
8.5	基于连接池和数据源访问数据库	225
8.5.1	创建 MySQL 数据库的连接池	227
8.5.2	创建数据源	230
8.5.3	基于数据源访问数据库	231
8.6	BLOB 字段的处理	232
8.7	分页显示	238
	小结	241
	习题 8	241
第 9 章	Java Mail 编程	242
9.1	常见电子邮件协议	242
9.2	Java Mail 基础	243
9.2.1	Java Mail 体系	243
9.2.2	Java Mail 主要对象	244
9.3	邮件服务器安装	246
9.3.1	安装邮件服务器	246
9.3.2	配置邮件服务器	247
9.3.3	测试邮件服务器	249
9.4	发送简单邮件	252
9.5	SMTP 身份认证	257
9.6	发送 HTML 邮件	260
9.7	接收邮件	264
9.8	处理附件	268
9.8.1	发送附件	268
9.8.2	接收附件	271

小结	275
习题 9	275
第 10 章 EJB 编程	276
10.1 EJB 基础知识	276
10.1.1 EJB 容器	276
10.1.2 EJB 组件	278
10.1.3 EJB 优点	279
10.1.4 EJB 分类	279
10.1.5 部署 EJB	279
10.2 无状态会话 Bean	280
10.2.1 什么是无状态会话 Bean	280
10.2.2 开发一个无状态会话 EJB	281
10.2.3 利用 Servlet 测试无状态会话 EJB	285
10.3 有状态会话 Bean	287
10.4 实体 Bean	294
10.4.1 什么是实体 Bean	294
10.4.2 持久化	295
10.4.3 创建持久性单元	295
10.4.4 创建实体 Bean “Notice”	297
10.4.5 利用 EntityManager 访问实体 Bean	299
10.5 消息驱动 Bean	303
小结	309
习题 10	309
第 11 章 Web 服务编程	310
11.1 Web 服务基础	310
11.2 什么是 Web 服务	311
11.3 Web 服务技术体系	311
11.4 Web 服务工作模型	313
11.5 Java EE 平台下的 Web 服务实现	313
11.6 开发 Web 服务实例	314
11.7 Web 服务的优缺点	321
小结	323
习题 11	323
第 12 章 综合练习	324
12.1 Java EE 体系架构基本知识	324
12.1.1 概述	324
12.1.2 架构模型	324
12.1.3 架构设计中的几点经验	326
12.2 项目背景介绍	326

12.3	数据库设计	326
12.4	系统整体架构	327
12.5	系统代码实现	328
12.6	系统部署	339
	小结	341
	参考文献	342

第 1 章 Java EE 概述

本章要点:

-
- Java EE 概念
 - Java EE 编程思想
 - Java EE 技术框架组成
 - 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 公司正式将 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。与简单的桌面应用程序相比较, 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个人计算机上, 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改变, 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。在二层体系结构中，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。

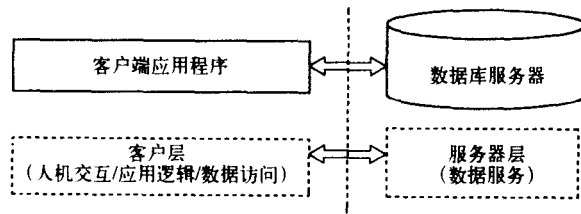


图 1-1 两层体系结构应用程序

这种体系结构应用程序有以下的缺点。

(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。

2. 三层体系结构应用程序

为解决二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、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作“瘦客户层”。数据服务层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作 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作“中间件服务器”或“中间件”，应用服务器层又被称作“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。

(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。

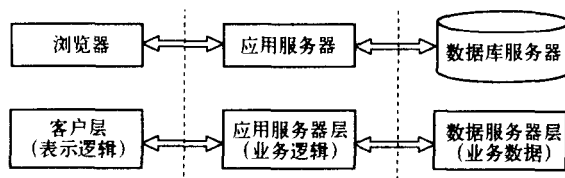


图 1-2 三层结构的应用程序

(3) 快速响应

通过中间件服务器层的负载均衡及缓存数据能力，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。

Java EE 客户层可以是网络浏览器也可以是桌面应用程序。

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）、Servlet、EJB（Enterprise JavaBeans）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。

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。

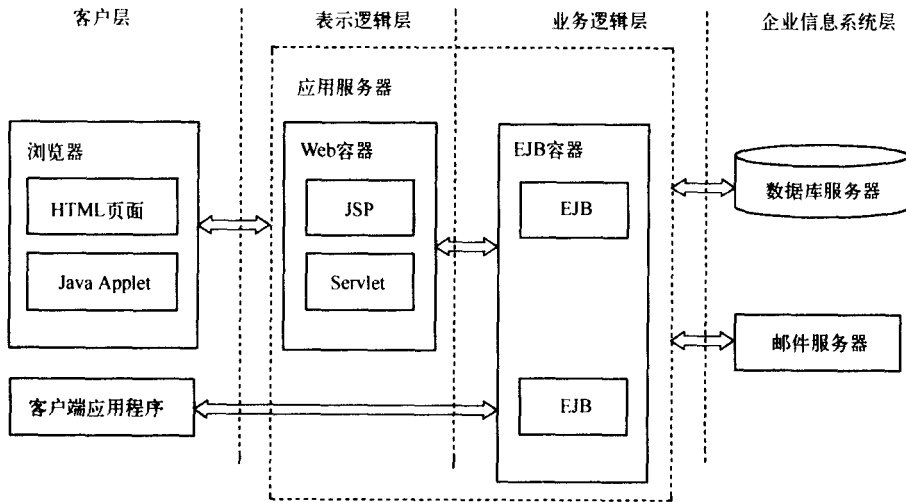


图 1-3 Java EE 多层体系结构

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的开发提供的标准，Sun 公司领导着 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://jcp.org/en/jsr/detail?id=244> 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—容器提供的服务的标准。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-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 JDBC（Java Data Base Connection, Java 数据库连接）、JMS（Java Message Service, Java 消息服务）等。

Java EE 规范已被众多中间件开发厂商接受并实现。Sun 公司开发了 Sun ONE 服务器，支持所有 Java EE 规范。BEA 公司用 WebLogic 实现了 Java EE 规范。IBM 公司的 WebSphere 也实现了 Java EE 规范，同时还有其他一些实现了 Java EE 规范的免费软件，如 JBoss 和 Tomcat 等。

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应用需求编程实现。

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组件除了要包含自身实现的代码文件外，还要包括一个 XML 文件，称为部署描述文件。部署描述文件中详细地描述了组件所要调用的容器服务的说明信息、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。

说明：在新推出的 Java EE 5 规范中，支持在组件的实现代码中引入标注信息来取代撰写复杂的部署描述文件，大大简化了 Java EE 应用程序的开发和部署。所谓的标注是一种 Java 修饰符，与 Java 代码中指定的 `public` 和 `private` 类似。关于标注，后面的章节中还会详细论述。

1.4 Java EE 技术框架

作为一个企业分布式应用开发标准，Java EE 最终由一系列的企业应用开发技术来实现。Java EE 技术框架可以分为三部分：组件技术、服务技术和通信技术。整个 Java EE 技术框架体系如图 1-4 所示。

1.4.1 组件技术

组件是 Java EE 应用的基本单元。Java EE 提供的组件主要包括三类：客户端组件、Web 组件和 EJB 组件。

1. 客户端组件

Java EE 客户端既可以是一个 Web 浏览器、一个 Applet，也可以是一个应用程序。

(1) Web 浏览器

Web 浏览器又称为瘦客户。它通常只进行简单的人机交互，不执行像查询数据库、执行复杂的业务规则等复杂操作。

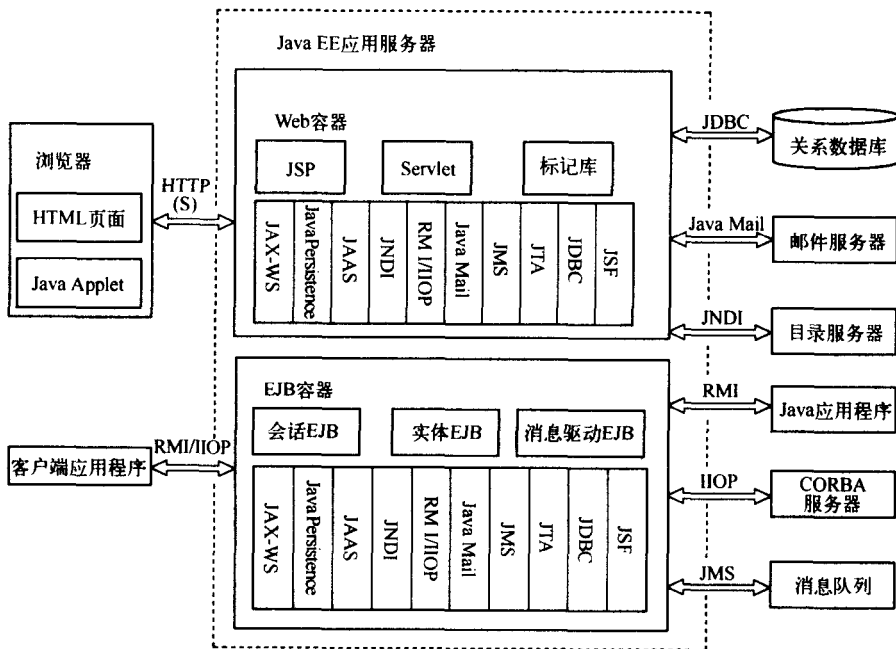


图 1-4 Java EE 技术体系结构

(2) Applet

Applet 是一个较小的用 Java 语言编写的程序，运行在浏览器上的虚拟机里，通过 HTTP 等协议与服务器进行通信。

(3) 应用程序客户端

Java EE 客户端应用程序运行在客户上，它为用户处理任务提供了比标记语言丰富的接口。典型的 Java EE 应用程序客户端拥有通过 Swing 和 AWT API 建立的图形用户界面。客户应用程序直接访问在服务器 EJB 容器内的 EJB 程序。当然，Java EE 客户应用程序也可像 Applet 客户那样以 HTTP 连接与服务器的 Servlet 通信。与 Applet 不同的是，客户应用程序一般需要在客户端进行安装，而 Applet 是通过 Web 下载，无须专门安装。

2. Web 组件

Web 组件是在 Java EE Web 容器上运行的软件程序。它的功能是在 HTTP 协议上对 Web 请求 (Request) 进行响应 (Response)。这些所谓响应其实是动态生成的网页。用户每次在浏览器上单击一个链接或图标，实际上是通过 Web 向服务器发出请求。Web 服务器负责将 Web 请求传递给 Web 组件。Java EE 平台的 Web 组件对这些请求进行处理后生成动态内容回复给客户端。