

- ◇以简练的语言讲解最新的开发技术
- ◇以精巧的示例演示最酷的编程技能
- ◇它带给你的不只是优秀程序员的必备技能，
更重要的是成为架构师思想和视野

Java EE

核心技术 与应用



郝玉龙 周旋◎著

Java EE

核心技术与应用

郝玉龙 周旋◎著



电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书基于最新的 Java EE 6 规范对 Java EE 应用开发技术进行系统讲解。书中主要包括四部分内容：第一部分介绍了 Java EE 的定义、设计思想、技术架构和开发模式等，可使读者全面认识 Java EE。第二部分以 Java EE 企业应用的表现层、数据持久化层和业务逻辑层的开发为主线，重点讲解 Java EE 6 规范的最新功能特性，包括 JSF 2.0、Servlet3.0、EJB 3.1 和 JPA 2.0 等，使读者掌握开发完整 Java EE 企业应用的基本技能。第三部分讲解了如何利用 CDI 实现组件的低耦合、如何利用 Bean Validation 框架实现统一校验、如何确保企业应用安全等高级知识，使读者掌握企业应用开发中的系统架构、安全防护等高级技能。第四部分讲解了如何利用 JavaMail、Web 服务、JMS 消息服务和消息驱动 Bean 等技术来实现企业应用之间的集成和交互。

本书可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及课时数量进行灵活调整。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Java EE 核心技术与应用 / 郝玉龙，周旋著. —北京：电子工业出版社，2013.5
ISBN 978-7-121-20175-2

I. ①J… II. ①郝… ②周… III. ①JAVA 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字（2013）第 074255 号

责任编辑：徐津平

印 刷：涿州市京南印刷厂

装 订：涿州市京南印刷厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：850×1168 1/16 印张：19.5 字数：524 千字

印 次：2013 年 5 月第 1 次印刷

印 数：4000 册 定价：45.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前言

为什么写作本书

企业应用的多用户、分布式、可扩展、安全性等高级特性使得开发企业级应用程序成为一项复杂而艰巨的任务。Java EE 是专为开发企业级应用而推出的标准规范和体系架构。自 1999 年 11 月推出以来, Java EE 经历了 5 次重大版本的升级。每一次的版本升级都会带给开发人员一些惊喜。但是 2009 年 11 月 Java EE 6 版本的推出却不能用“惊喜”二字来表达, 它给 Java EE 开发领域带来的是一次重大的革命, 而不是简单的改进。为了帮助广大读者尽快掌握最新的 Java EE 技术, 特推出此书。让我们一起分享新的 Java EE 开发技术带给我们的快乐!

什么人适合读这本书

如果你已经学习了 Java 语言, 准备开始学习 Java EE 应用开发, 那么这本书正是你在寻找的。它将带你一起系统学习 Java EE 最新的开发技术。本书由浅入深、全面深入地讲解了 Java EE 应用开发的各个环节所必需的基本技能和实战要领, 你最终收获的将不仅是 Java EE 应用开发技能的掌握, 更重要的是对 Java EE 编程思想的理解, 及对 Java EE 经典架构模式的领悟。

如果你已经是一名 Java EE 应用开发人员, 正醉心于 Struts、Spring 和 Hibernate 等开源框架的海洋里, 那么是时候升级到最新的 Java EE 技术了。新的 Java EE 开发技术将使你以一种更加清晰的体系架构、更加优雅的系统设计、更加简洁的编程模型来完成任务。这本书将是你快速升级的最好阶梯。它涵盖了 Java EE 最新的功能特性, 展示了 Java EE 新的编程思想、架构模式, 昭示了 Java EE 开发未来的发展趋势。书中专门针对经验开发者之前掌握的开发技术与新的 Java EE 技术进行了对比分析, 便于加深对新技术的理解。

本书有哪些特点

(1) 体系完整, 内容新颖

Java EE 是一个复杂的技术体系, 本书对 Java EE 进行了整体介绍, 涵盖 Java EE 体系架构的各个层次及其相关的功能特性, 其中的 JSF 扩展、CDI、Bean Validation、安全、Restful Web 服务等重要内容都是市面上其他图书尚未涉及的, 可使读者获得对 Java EE 应用开发技术的全面理解。

(2) 重点突出, 言简意赅

针对企业应用开发过程中的常见任务, 本书对最常用的 JSF、JPA 和 EJB 等内容, 以通俗简练的语言进行了深入细致的讲解, 重点讲述核心概念和开发流程, 突出体现 Java EE 设计思想和架构模式。

(3) 结构严谨, 由浅入深

为降低学习难度, 使得读者可循序渐进地掌握 Java EE 开发技术, 书中内容按照基础技术、高级特性和应用整合三个梯次由浅入深展开。基础技术部分以企业应用的表现层、数据持久化层和业务逻辑层的开发为主线, 重点讲解 Java EE 6 规范的最新功能特性, 包括 JSF 2.0、Servlet3.0、EJB 3.1 和

JPA 2.0 等；高级特性部分讲解了企业应用开发中的系统架构、安全防护、数据校验等高级技能；应用集成部分讲解了如何利用 JavaMail、Web 服务、JMS 消息服务等内容来实现企业应用之间的集成和交互。

(4) 示例简洁，举一反三

本人根据多年 Java EE 工程项目实践与实际教学经验，精选示例来演示开发技术。示例力求简洁，便于读者理解和操作。对示例重在讲解启发，而不是简单的代码堆砌。

致谢

本书编写过程中，得到了许多人的帮助和支持。感谢我的同事潘老师、关老师和姚老师，他们对本书的定位和内容组织提出了宝贵的建议。感谢周旋、季平和胡志宇等同事，他们参与了本书的部分编写工作。感谢我的研究生们，他们测试了本书的全部代码。感谢本书的编辑，她对本书尽快出版付出了艰辛的劳动。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。

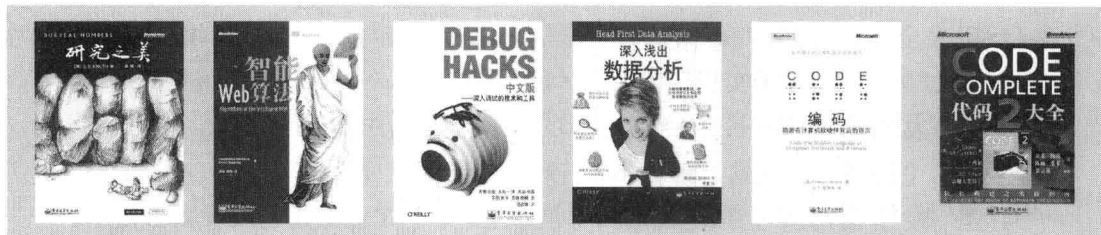
由于作者水平有限，加之编写时间仓促，书中难免出现错误和不足。对于书中的任何问题，请发 E-mail 至邮箱：haoyulongsd@163.com。

郝玉龙

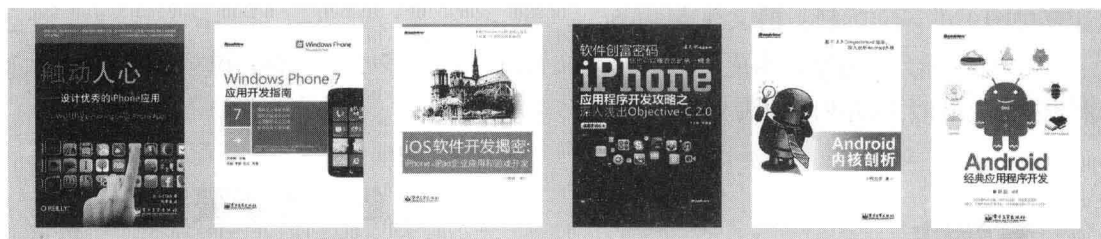
2012 年 11 月于北京

博文视点精品图书展台

专业典藏



移动开发



物联网

云计算



数据库

Web开发



程序设计



办公精品

网络营销



博文视点诚邀精锐作者加盟

九载耕耘奠定专业地位

以书为证彰显卓越品质

《代码大全》、《Windows内核情景分析》、《加密与解密》、《编程之美》、《VC++深入详解》、《SEO实战密码》、《PPT演义》……

“圣经”级图书光耀夺目,被无数读者朋友奉为案头手册传世经典。

潘爱民、毛德操、张亚勤、张宏江、咎辉Zac、李刚、曹江华……

“明星”级作者济济一堂,他们的名字熠熠生辉,与IT业的蓬勃发展紧密相连。

九年的开拓、探索和励精图治,成就博古通今、文圆质方、视角独特、点石成金之计算机图书的风向标杆:博文视点。

“凤翱翔于千仞兮,非梧不栖”,博文视点欢迎更多才华横溢、锐意创新的作者朋友加盟,与大师并列于IT专业出版之巅。

英雄帖

江湖风云起,代有才人出。

IT界群雄并起,逐鹿中原。

博文视点诚邀天下技术英豪加入,

指点江山,激扬文字

传播信息技术,分享IT心得

专业的作者服务

博文视点自成立以来一直专注于IT专业技术图书的出版,拥有丰富的与技术图书作者合作的经验,并参照IT技术图书的特点,打造了一支高效运转、富有服务意识的编辑出版团队。我们始终坚持:

善待作者——我们会把出版流程整理得清晰简明,为作者提供优厚的稿酬服务,解除作者的顾虑,安心写作,展现出最好的作品。

尊重作者——我们尊重每一位作者的技术实力和生活习惯,并会参照作者实际的工作、生活节奏,量身制定写作计划,确保合作顺利进行。

提升作者——我们打造精品图书,更要打造知名作者。博文视点致力于通过图书提升作者的个人品牌和技术影响力,为作者的事业开拓带来更多的机会。



联系我们

博文视点官网: <http://www.broadview.com.cn>

新浪官方微博: <http://weibo.com/broadviewbj>

投稿电话: 010-51606000-3254368

CSDN官方博客: <http://blog.csdn.net/broadview2006/>

腾讯官方微博: <http://t.qq.com/bowenshidian>

投稿邮箱: jsj@phei.com.cn

目 录

第 1 章 走进 Java EE	1
1.1 引言	1
1.2 为什么需要 Java EE	1
1.2.1 企业级应用特征	1
1.2.2 企业级应用架构体系	2
1.3 什么是 Java EE	3
1.4 Java EE 设计思想	4
1.4.1 容器	4
1.4.2 组件	5
1.4.3 容器与组件的交互	5
1.5 Java EE 技术架构	5
1.5.1 组件技术	5
1.5.2 服务技术	7
1.5.3 通信技术	8
1.5.4 框架技术	9
1.6 Java EE 核心开发模式	9
1.7 Java EE 优点	10
1.8 小结	11
第 2 章 搭建开发环境	12
2.1 引言	12
2.2 安装 JDK	12
2.3 安装 NetBeans IDE	13
2.4 测试开发环境	14
2.5 小结	16
第 3 章 基于 JSF 构建表示逻辑层	17
3.1 引言	17
3.2 什么是 JSF	17
3.2.1 什么是框架	17
3.2.2 为什么需要框架	17
3.2.3 JSF 是什么样的框架	18
3.2.4 为什么学习 JSF	19
3.3 第一个 JSF 应用	19
3.3.1 创建 JSF 项目	19

3.3.2	模型组件	21
3.3.3	视图组件	22
3.3.4	控制组件	22
3.3.5	运行演示	23
3.3.6	总结思考	23
3.4	利用 JSF 组件构建视图	24
3.4.1	JSF 标记库	24
3.4.2	HTML 标记	25
3.4.3	Core 标记	39
3.4.4	使用 JSTL 标记	40
3.5	在视图中访问 Web 资源	41
3.6	利用页面模板提高视图可维护性	43
3.6.1	布局	43
3.6.2	装饰	48
3.7	利用 Managed Bean 封装业务逻辑	51
3.7.1	定义 Managed Bean	51
3.7.2	生命周期范围	53
3.7.3	Bean 之间的依赖	56
3.7.4	生命周期回调方法	58
3.8	使用 EL 访问服务器端数据	59
3.8.1	范围	59
3.8.2	访问对象	59
3.8.3	值表达式和方法表达式	60
3.8.4	延迟计算	61
3.9	实现灵活的导航控制	61
3.9.1	视图 ID	61
3.9.2	利用 Post 请求实现导航	62
3.9.3	导航约定	66
3.9.4	导航规则	67
3.9.5	重定向	69
3.9.6	利用 Get 请求实现导航	71
3.9.7	JSF 框架外导航	74
3.9.8	导航中的参数传递	74
3.9.9	导航总结	77
3.10	实现国际化支持	77
3.10.1	准备资源包	77
3.10.2	配置资源包	79
3.10.3	在 JSF 视图中使用资源	79
3.10.4	设置应用程序本地属性	80
3.11	使用 Ajax 获得更好的用户体验	81
3.12	小结	83

第 4 章 扩展 JSF	84
4.1 引言	84
4.2 JSF 请求处理过程	84
4.2.1 常规流程	84
4.2.2 示例分析	85
4.2.3 特殊流程	88
4.2.4 异常处理	89
4.2.5 总结思考	92
4.3 利用监听器实现事件处理	93
4.3.1 Value Change 事件	93
4.3.2 Action 事件	96
4.3.3 Phase 事件	98
4.3.4 System 事件	99
4.4 自定义类型转换	101
4.4.1 标准转换器	102
4.4.2 自定义转换器	102
4.5 自定义输入校验	105
4.5.1 标准校验器	105
4.5.2 自定义校验器	106
4.5.3 Bean 方法校验	107
4.5.4 异常信息本地化	108
4.6 自定义复合组件	108
4.6.1 复合组件标记库	108
4.6.2 定制简单的复合组件	109
4.6.3 开发复杂的复合组件	111
4.7 自定义非 UI 组件	116
4.8 自定义 UI 组件	119
4.8.1 创建一个简单的 UI 组件	119
4.8.2 利用属性控制自定义组件行为	121
4.8.3 使用单独的渲染器	123
4.8.4 获取用户输入信息	125
4.8.5 保存组件状态	127
4.9 使用第三方组件	129
4.10 小结	130
第 5 章 利用 Servlet 处理复杂 Web 请求	131
5.1 引言	131
5.2 Servlet 基础	131
5.2.1 什么是 Servlet	131
5.2.2 Servlet 工作流程	131
5.2.3 Servlet API	132

5.3	第一个 Servlet	133
5.4	处理请求	138
5.5	生成响应	141
5.6	在 JSF 应用中处理非 JSF 请求	142
5.7	支撑自定义 JSF 组件	146
5.8	利用 Filter 过滤请求	149
5.9	小结	152
第 6 章	利用 JPA 访问企业信息	153
6.1	引言	153
6.2	数据库驱动与 JDBC	153
6.3	连接池和数据源	154
6.3.1	基本概念	154
6.3.2	创建 MySQL 连接池	156
6.3.3	创建数据源	157
6.4	第一个 JPA 应用	158
6.4.1	持久化单元	158
6.4.2	Entity	159
6.4.3	EntityManager	161
6.4.4	运行演示	162
6.5	ORM	163
6.5.1	Entity	163
6.5.2	主键	164
6.5.3	复合主键	164
6.5.4	属性	167
6.5.5	关联映射	170
6.5.6	加载方式	173
6.5.7	顺序	174
6.5.8	继承映射	174
6.6	Entity 管理	175
6.6.1	获取 EntityManager	175
6.6.2	持久化上下文	176
6.6.3	Entity 操作	176
6.6.4	级联操作	183
6.7	JPQL	184
6.7.1	动态查询	185
6.7.2	参数设置	186
6.7.3	命名查询	186
6.7.4	属性查询	187
6.7.5	使用构造器	187
6.8	基于 Criteria API 的安全查询	188
6.9	缓存	190

6.10	并发控制	192
6.11	生命周期回调方法	194
6.12	小结	195
第 7 章	使用会话 Bean 实现业务逻辑	196
7.1	引言	196
7.2	EJB 基础	196
7.2.1	为什么需要 EJB	196
7.2.2	EJB 容器	197
7.2.3	EJB 组件	198
7.2.4	EJB 接口	199
7.2.5	EJB 分类	199
7.2.6	部署 EJB	200
7.2.7	EJB 优点	200
7.3	无状态会话 Bean	200
7.3.1	什么是无状态会话 Bean	201
7.3.2	开发一个无状态会话 Bean	202
7.3.3	利用 Servlet 测试无状态会话 Bean	205
7.3.4	利用远程客户端测试无状态会话 Bean	207
7.4	有状态会话 Bean	208
7.4.1	基本原理	208
7.4.2	实现有状态会话 Bean	209
7.5	单例会话 Bean	213
7.5.1	基本原理	213
7.5.2	利用 JSF 访问单例会话 Bean	213
7.5.3	并发控制	215
7.5.4	依赖管理	216
7.6	Time 服务	217
7.7	拦截器	219
7.8	事务支持	221
7.9	异步方法	222
7.10	小结	227
第 8 章	利用 CDI 实现组件间低耦合	228
8.1	引言	228
8.2	CDI 概述	228
8.3	CDI 下的受控 Bean	228
8.4	Bean 的生命周期范围	232
8.5	使用限定符注入动态类型	237
8.6	使用替代组件实现部署时动态注入	239
8.7	使用生产方法注入动态内容	241
8.8	使用拦截器绑定类型注入功能服务	243

8.9	利用构造型封装注入操作	246
8.10	小结	247
第 9 章	使用 Bean Validation 校验数据	248
9.1	引言	248
9.2	Bean Validation 概述	248
9.3	使用默认约束器	248
9.4	实现自定义约束器	251
9.5	约束的传递	254
9.5.1	继承	254
9.5.2	级联	255
9.6	小结	256
第 10 章	确保企业应用安全	257
10.1	引言	257
10.2	认证	257
10.2.1	配置文件安全域	257
10.2.2	配置 JDBC 安全域	258
10.2.3	声明认证配置	260
10.3	授权	260
10.3.1	授权声明	260
10.3.2	角色映射	261
10.4	测试 Java EE 容器的安全服务	262
10.5	定制 FORM 方式认证界面	262
10.6	在代码中获取用户身份信息	264
10.7	EJB 安全控制	265
10.8	小结	267
第 11 章	为应用添加邮件发送功能	268
11.1	引言	268
11.2	JavaMail 基础	268
11.2.1	JavaMail 体系	268
11.2.2	JavaMail 规范组成	269
11.3	配置 JavaMail 会话	270
11.4	发送邮件	270
11.5	发送带附件的邮件	272
11.6	小结	275
第 12 章	利用 Web 服务集成应用	276
12.1	引言	276
12.2	Web 服务概述	276
12.2.1	什么是 Web 服务	276

12.2.2	Web 服务技术体系	276
12.2.3	Web 服务工作模型	278
12.3	Java EE 平台下的 Web 服务实现	278
12.4	开发 Web 服务实例	278
12.4.1	创建 Web 服务组件	279
12.4.2	为 Web 服务组件添加业务逻辑	280
12.4.3	部署 Web 服务	280
12.4.4	测试 Web 服务	280
12.5	调用 Web 服务	282
12.5.1	添加 Web 服务客户端	282
12.5.2	调用 Web 服务	283
12.6	将会话 Bean 发布为 Web 服务	285
12.7	RESTful Web 服务	286
12.7.1	什么是 REST	286
12.7.2	利用 JAX-RS 开发 RESTful Web 服务	287
12.7.3	与 SOAP 对比	290
12.8	Web 服务的优缺点	291
12.9	小结	292
第 13 章	利用消息服务实现应用间异步交互	293
13.1	引言	293
13.2	JMS 概述	293
13.2.1	JMS 消息模型	293
13.2.2	JMS 消息服务接口	293
13.2.3	消息传递模式	294
13.3	配置消息服务资源和连接工厂	294
13.4	发送 JMS 消息	295
13.5	利用 MDB 处理消息	296
13.6	小结	298

第 1 章 走进 Java EE

1.1 引言

在开始学习 Java EE 之前,我们首先要了解什么是 Java EE?简单来说,Java EE 是一个用来开发企业级应用的架构体系和标准规范集合。但显然这不是一个能令人满意的回答。你可能马上会问:什么是企业级应用?什么又是架构体系?Java EE 是一个怎样的架构体系?它都包含了哪些标准规范呢?在本章我们将详细讲述 Java EE 产生的背景、组成、设计思想和技术架构等,帮你概要了解 Java EE 的整体特性,为进一步学习 Java EE 开发技术打下基础。

1.2 为什么需要 Java EE

1.2.1 企业级应用特征

Java EE 是为了满足企业级应用开发而推出的。大部分读者可能都开发过应用程序,例如老师布置的作业或一些小的工具软件,但是这些应用程序与企业级应用程序有着本质的区别。企业级应用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、为大量用户提供连续服务的应用程序。如网上电子银行就是一个典型的企业应用。与常见的应用程序相比,企业级应用一般具有以下特点。

- (1) 多用户。企业级应用通常需要服务大量用户群体,少则是一个单位或组织内的几十名员工,多则是数以亿计的社会人群。
- (2) 分布式。企业级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。
- (3) 连续性。企业级应用通常需要 24×7 连续不停的运转,即使是短暂的服务中断也可能是无法接受的,例如铁路调度系统、电子商务网站等。
- (4) 多变性。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。企业级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。
- (5) 可扩展性。在网络环境内,应用的潜在用户可能成百上千,企业级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。
- (6) 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,还必须保证企业信息的安全和系统运行的可靠性。
- (7) 集成化。企业应用除了满足自身的需求外,还经常需要与其他信息系统进行交互对接。例如一个电子商务网站通常需要与物流信息系统和电子支付系统进行交互。

注: Java EE 是专为解决企业级应用开发提出的,牢记企业应用的上述特性是深入理解和灵活运用 Java EE 开发技术的前提和基础。

1.2.2 企业级应用架构体系

应用程序的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分为客户层（Client）和服务层（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。

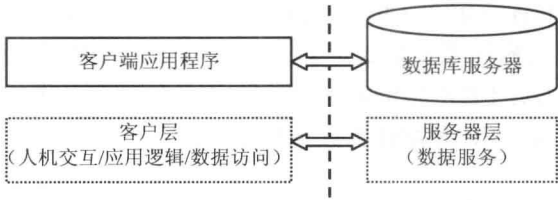


图 1-1 两层体系结构应用程序

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

为了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，在两层体系结构应用程序的客户层与服务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。与两层体系结构应用程序相比，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”。数据服务层仍旧仅提供数据信息服务。由于客户层应用程序通常是由一个通用的浏览器（Browser）程序来实现的，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”。

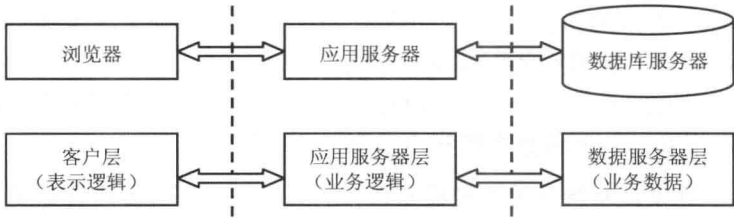


图 1-2 三层体系结构应用程序

与两层体系结构的应用程序相比，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。

(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。多层体系结构应用程序与三层体系结构类似，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。

1.3 什么是 Java EE

1. Java EE 是一个企业级应用架构体系

不要被名称“Java Platform Enterprise Edition”误导，与 Java 不同，Java EE 是一个企业级应用的架构体系，而不是一门编程语言。Java 作为一门编程语言，可以用来编写各种应用程序。而 Java EE 作为一个架构体系，它定义了企业级应用的层次结构，旨在简化和规范企业应用系统的开发和部署。

典型的 Java EE 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层，如图 1-3 所示。

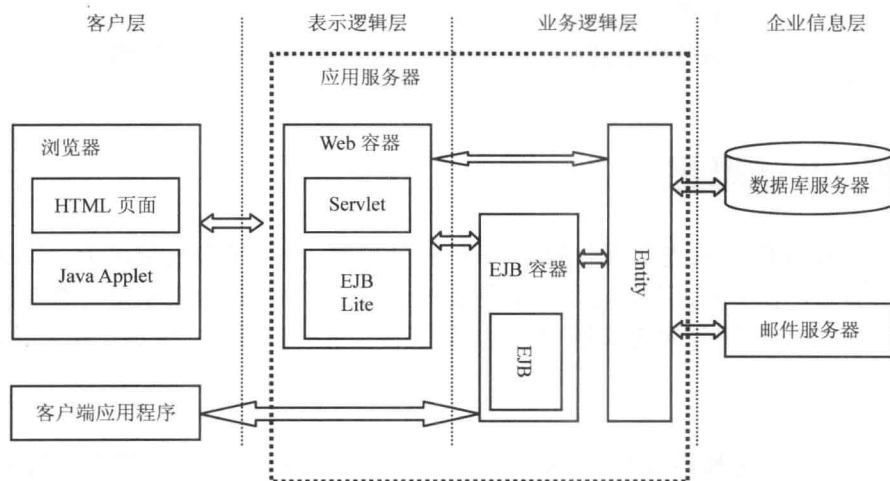


图 1-3 Java EE 多层体系结构

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）和业务逻辑层位于应用服务器上，都是由一些 Java EE 标准组件来实现，这些组件运行在兼容 Java EE 标准的应用服务器上，来实现特定的表现逻辑和业务逻辑。由于企业级