

新工科建设之路·软件工程规划教材



Java EE 企业级应用技术



姜志强 编著

 中国工信出版集团

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

新工科建设之路·软件工程规划教材

Java EE 企业级 应用技术

姜志强 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一本讲授 Java EE 企业级应用技术的基本概念、基本框架和程序设计基本方法的教材。

全书共 8 章,第 1 章介绍 Java EE 企业级应用的基本知识和基本概念、企业级应用与中间件技术、Java EE 体系架构的基本模式;第 2 章详细讲解 Java Server Faces 框架的主要概念、主要组件和主要技术构成;第 3 章讲解上下文、资源注入和依赖注入的概念;第 4 章讲解 Java EE 体系中的核心内容之一企业 Bean 的基本概念,包括有状态会话 Bean、无状态会话 Bean、单身会话 Bean、消息服务与消息驱动 Bean 等几种企业 Bean 类型;第 5 章专门介绍 Java 持久性与事务的相关概念和知识;第 6 章介绍 Web 服务及相关的基本概念,包括用 JAX-WS 技术构建 Web 服务和用 JAX-RS 技术构建 RESTful Web 服务;第 7 章简要讨论安全性的基本概念;第 8 章简单介绍 Java 消息服务和 Java EE 拦截器技术。

本书以 Java EE 5、Java EE 6、Java EE 7 技术规范为蓝本,重点介绍 B/S 体系结构模式下多层应用体系结构的最新开发技术——JSF+EJB+JPA 技术组合的使用和开发。

本书适合作为普通高等院校计算机科学与技术、软件工程及相关专业课程的教材,也可供其他专业的本科生、研究生及各级计算机专业技术人员参考使用。

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 企业级应用技术/姜志强编著. —北京:电子工业出版社,2019.1

ISBN 978-7-121-34444-2

I. ①J… II. ①姜… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 124264 号

策划编辑:章海涛

责任编辑:底波

印刷:三河市君旺印务有限公司

装订:三河市君旺印务有限公司

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:787×1092 1/16 印张:12 字数:307.2 千字

版次:2019 年 1 月第 1 版

印次:2019 年 1 月第 1 次印刷

定价:39.00 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, 88258888。

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。

本书咨询联系方式:192910558 (QQ 群)。

前 言

Java 语言自从 1995 年发布，至今已经有 20 多年的发展历程，早已成为重要的软件开发工具，成为信息行业占有率很高的开发设计平台。Java 平台分为标准版（Java SE）、企业版（Java EE）和微型版（Java ME）。实际上，Java 语言和 Java 技术如此高的市场占有率，绝大部分缘于其企业版的市场占有率。可以说，真正占有信息行业市场的，是其企业开发设计平台。

Java 企业级开发技术自 1999 年 Java 1.2 版发布至今，已经成为一种专门为互联网设计的与平台无关的网络分布式开发设计技术，广泛应用于各种信息平台的开发设计，既是一种开发工具，也是一种企业级技术标准。在电信、移动技术、银行、证券、电子商务等领域，都是主流的开发设计技术，在开发设计中占优势和主导地位。目前大部分的大型信息化平台都选择 Java 企业级技术来构建。Java EE 技术的发展经历了两个明显的历史发展阶段。第一阶段是 1999 年的 J2EE 1.2 版到 2003 年的 J2EE 1.4 版，其主干技术框架为 JSP、Servlet、EJB、JDBC，同时还有第三方的框架产品 Struts、Spring、Hibernate 可以作为辅助开发框架；第二阶段是 2006 年的 Java EE 5 版到 2013 年的 Java EE 7 版，其主干技术框架为 JSF、EJB、JPA，其中吸收了很多第三方产品的设计理念和技术内容。早期的 JSP、Servlet、EJB、JDBC 是一套已经十分成熟的架构，业内的资料也十分丰富；而采用了 JSF、EJB、JPA 技术规范组合的内容却是能够真正反映 Java 技术的最新成就的内容。

本书的主旨就是介绍 JSF、EJB、JPA 等技术规范及其组合，讲授新的技术框架。

吉林大学软件学院是国家首批“国家级示范性软件学院”，自 2003 年开始招收全日制本科生，从 2003 年第一届学生开始，就开设了讲授 Java 企业级技术的课程。2013 年，该学院的“软件工程”专业被正式列入《卓越工程师教育培养计划第三批学科专业名单》的本科专业目录。为深入贯彻落实国家卓越工程师教育培养计划，该学院专门成立“软件工程师卓越班”，每年单独选拔招收学生 30 名左右，并为这个教学班单独设立教学计划，在软件工程专业正常教学内容的基础上，课程的侧重点力求能够更倾向于教育部和中国工程院专门制定的《卓越工程师教育培养计划通用标准》中所要求的要点。“Java EE 企业级应用技术”课程被列入“软件工程师卓越班”的教学计划，作为一门专业必修课，也是该学院重点建设的一门理论联系实际性质的专业课。2016 年《Java EE 企业级应用技术》教材建设被列入“吉林大学本科‘十三五’规划教材”建设计划。

本书的内容是近十几年课程教学过程的积累，也是近几年对 Java EE 企业级技术的消化和学习的结果。实际上课程的教学内容随着 Java 技术的不断发展，也经过了几次调整，更新了很多内容，既有摒弃，也有积淀。本书主要介绍 Java EE 第 5 版到第 7 版的主要体系框架，力求能够反映其中的基本概念、基本思想、基本框架。由于 Java EE 是一个庞大的技术体系，无法面面俱到地介绍，因此在内容上尽量以 JSF、EJB、JPA 技术为重点，力图使学生经过教学过程之后能够掌握最基本的框架，对 Internet 环境下的分布式系统开发和 Java EE 平台有一个基本的理解。

本书建议理论授课学时为 32 学时，建议实践教学学时为 16 学时，具体的学时分配数在

每章都会给出，可以根据教学侧重点进行增减。实践教学环节的题目和内容设置，任课教师可以根据学生的技术水平灵活掌握。

在本书即将付梓之际，我要感谢吉林大学对本书的大力支持，在“十三五”规划教材申报和答辩过程中，学校专家组的专家们对我和本书的鼓励和肯定依然使我记忆犹新。我想用这本书告慰逝去的父亲，父亲对儿子的殷切期望是我克服写作过程中的怠惰情绪、克服心智和体力上的困苦与疲惫、完成写作的重要动力。能够在父亲离世三周年之前完成此书，了却了我的心愿。这也是我继 2007 年出版《Java 语言程序设计》之后，再一次用自己的劳动成果向父亲汇报。我还要把这本书作为礼物送给我的女儿，希望父女之间互相鼓励、共同进步，彼此践行承诺。

在本书的最后一个字符录入完成之际，我知道，出版社的编辑们及各个环节的工作人员就要开始辛苦忙碌了。没有他们的辛勤付出，没有他们高质量和卓有成效的工作，一本书的出版是不可能的。与他们合作的过程，也是我与他们分享工作乐趣的过程。

由于时间仓促，加之水平有限，对于新技术规范的理解和消化也还不够透彻，书中难免存在一些缺点和错误，恳请专家和读者批评指正。

姜志强
于吉林大学

目 录

第 1 章	Java EE 企业级应用概述	1
1.1	企业级应用与中间件	1
1.1.1	什么是企业级应用	1
1.1.2	当今的计算机应用环境	1
1.1.3	Internet 时代计算机应用的主要矛盾	2
1.1.4	中间件的概念	2
1.1.5	中间件的发展	3
1.2	Web 应用的发展	4
1.2.1	静态网页技术	4
1.2.2	动态网页技术	5
1.2.3	C/S 体系结构模式与 B/S 体系结构模式	6
1.2.4	多层应用体系结构	7
1.3	Java EE 概述	8
1.3.1	Java EE 模式的发展	8
1.3.2	Java EE 应用模式	9
1.3.3	Java EE 组件与容器	10
1.3.4	Java EE 体系架构的优越性	11
1.3.5	Java EE API	12
1.4	主要 Java EE 产品介绍	13
1.4.1	WebSphere	13
1.4.2	WebLogic	13
1.4.3	GlassFish	13
1.4.4	Tomcat	14
1.4.5	JBoss	14
1.5	用 GlassFish 服务器构建企业级应用环境	14
1.5.1	下载正版安装软件	14
1.5.2	安装 Java 软件和 GlassFish 服务器	15
1.5.3	使用 Java 软件和 GlassFish 服务器	19
第 2 章	Web 应用框架 JSF	21
2.1	Web 应用概述	21
2.1.1	Web 应用的基本概念	21
2.1.2	Web 应用的工作过程	21
2.1.3	Web 应用的基本开发步骤	22
2.2	JavaServer Faces 框架技术	22

2.2.1	JavaServer Faces 框架的概念	22
2.2.2	生成一个简单的 JavaServer Faces 框架的 Web 应用	23
2.2.3	JavaServer Faces 框架的体系结构	27
2.2.4	JavaServer Faces 框架的 Web 应用的生命周期	28
2.3	XHTML 规范	29
2.3.1	认识 XHTML 规范	29
2.3.2	XHTML 语法规则与 HTML 语法规则的简单比较	29
2.3.3	XHTML 的页面结构	30
2.3.4	现行 XHTML 规范	31
2.4	表达式语言	32
2.4.1	什么是表达式语言	32
2.4.2	即刻求值表达式和延缓求值表达式	32
2.4.3	值表达式和方法表达式	33
2.4.4	表达式语言语法	35
2.5	UI 标签组件技术	36
2.5.1	JavaServer Faces 标签库及组件 API	36
2.5.2	创建 JSF 页面和使用 UI 组件	36
2.5.3	生成用户 UI 组件	41
2.6	Backing Bean	44
2.6.1	什么是 Backing Bean	44
2.6.2	Backing Bean 中的属性	44
2.6.3	Backing Bean 中的方法	45
2.7	导航	45
2.7.1	什么是导航	45
2.7.2	部署描述符文件和应用配置资源文件	45
2.7.3	静态导航	47
2.7.4	动态导航	47
2.8	JavaServer Faces 事件处理机制	48
2.8.1	JavaServer Faces 的事件与事件处理	48
2.8.2	事件与监听器 API	48
2.8.3	实现监听器的两种方式	48
2.8.4	在组件上注册监听器	49
2.9	转换器	50
2.9.1	转换器的概念	50
2.9.2	标准转换器	50
2.9.3	注册、使用转换器	51
2.9.4	自定义转换器	53
2.10	验证器	54

2.10.1	验证器的概念	54
2.10.2	标准验证器	54
2.10.3	注册、使用验证器	55
2.10.4	自定义验证器	56
2.10.5	一个使用了监听器、转换器和验证器的完整例子	56
2.11	Facelets 与复合组件	62
2.11.1	什么是 Facelets	62
2.11.2	开发一个简单的 Facelets 应用	62
2.11.3	模板	62
2.11.4	复合组件	63
2.11.5	应用程序的目录结构与资源	64
2.12	Servlet	64
2.12.1	Servlet 的基本概念	64
2.12.2	编写 Servlet 程序	66
2.12.3	使用 Servlet	67
第 3 章	上下文和注入	70
3.1	上下文和注入的概念	70
3.1.1	什么是上下文和注入	70
3.1.2	托管 Bean	72
3.2	依赖注入	73
3.2.1	创建一个支持依赖注入的项目	73
3.2.2	用依赖注入的方式注入 Bean	74
3.2.3	用生产者方法注入对象	78
3.2.4	配置一个 CDI 应用	79
3.3	资源连接和资源注入	79
3.3.1	资源与 JNDI 命名	79
3.3.2	资源注入	80
第 4 章	企业 Bean	81
4.1	企业 Bean 概述	81
4.1.1	什么是企业 Bean	81
4.1.2	企业 Bean 的发展	81
4.1.3	企业 Bean 的类型	82
4.1.4	开发企业 Bean 的基本要求	82
4.2	会话 Bean	85
4.2.1	什么是会话 Bean	85
4.2.2	访问会话 Bean 的几种方式	85
4.2.3	有状态会话 Bean	87
4.2.4	无状态会话 Bean	93

4.2.5	单身会话 Bean	96
4.3	消息驱动 Bean	99
4.3.1	什么是消息驱动 Bean	99
4.3.2	消息驱动 Bean 与 Java 消息服务	103
4.4	企业 Bean 高级技术	103
4.4.1	使用嵌入式企业 Bean 容器	103
4.4.2	在会话 Bean 中使用异步方法调用	103
第 5 章	持久性与事务	105
5.1	在 Java EE 环境中使用数据库	105
5.1.1	对象关系映射的概念	105
5.1.2	在 Java EE 中使用数据库	106
5.2	持久性与 Java 持久性 API	106
5.2.1	实体和实体类	106
5.2.2	实体之间的关联关系	112
5.2.3	实体的继承层次	114
5.2.4	实体的管理和操作	114
5.2.5	查询实体	117
5.3	Java 持久性查询语言	117
5.3.1	查询语言术语	117
5.3.2	用 Java 持久性查询语言生成查询	117
5.3.3	Java 持久性查询语言的基本语句	119
5.4	标准 API	120
5.4.1	标准 API 与中间模型 API	120
5.4.2	使用中间模型 API 翻模实体类	121
5.4.3	使用标准 API 与中间模型 API 查询	122
5.5	事务	125
5.5.1	事务的概念	125
5.5.2	Java 事务 API	126
5.5.3	一个使用持久性和 Java 事务 API 的实例	127
第 6 章	Web 服务	135
6.1	Web 服务的概念	135
6.1.1	什么是 Web 服务	135
6.1.2	JAX-WS Web 服务与 JAX-RS Web 服务	135
6.1.3	确定使用哪种类型的 Web 服务	136
6.2	SOAP 与 WSDL 简介	136
6.2.1	SOAP 简介	136
6.2.2	WSDL 简介	141
6.3	用 JAX-WS 构建 Web 服务	142

6.3.1	JAX-WS 简述	142
6.3.2	用 JAX-WS 生成一个简单的 Web 服务	142
6.3.3	一个简单的 JAX-WS Application 客户端	144
6.3.4	一个简单的 JAX-WS Web 客户端	144
6.4	用 JAX-RS 构建 RESTful Web 服务	146
6.4.1	什么是 RESTful Web 服务	146
6.4.2	用 JAX-RS 开发 RESTful Web 服务	147
第 7 章	安全性	149
7.1	安全性概述	149
7.1.1	Java EE 安全性概述	149
7.1.2	安全机制	150
7.1.3	安全容器	151
7.1.4	领地、用户、组群和角色	151
7.2	Web 应用安全的例子	152
7.2.1	一个声明性安全的例子	152
7.2.2	一个编程性安全的例子	154
7.3	企业应用安全的例子	156
7.3.1	一个声明性安全的例子	156
7.3.2	一个编程性安全的例子	158
第 8 章	Java EE 支持技术	160
8.1	消息服务	160
8.1.1	消息服务的概念	160
8.1.2	JMS API	161
8.1.3	JMS 消息发送和接收实例	164
8.2	Java EE 拦截器	171
8.2.1	拦截器的概念	171
8.2.2	使用拦截器	172
附录 A	Java 持久性查询语言语法的 Backus-Naur Form 表述	176
	参考文献	182

第 1 章 Java EE 企业级应用概述

本章主要内容：Java EE 企业级应用技术是目前分布式应用的主流开发技术。本章从企业级应用的概念开始，逐步介绍企业级应用与中间件技术，在此基础上介绍 Web 应用的发展历程，从而引入 Java EE 技术。接着介绍 Java EE 企业级应用技术的发展、Java EE 体系架构的基本模式、Java EE 的组件与容器、Java EE 体系架构的优越性，简单介绍几款主要的 Java EE 产品，最后介绍用 Java SE 平台、GlassFish 服务器和 NetBeans IDE 构建 Java EE 企业级应用环境的基本操作步骤。

建议讲授课时数：4 课时。

1.1 企业级应用与中间件

1.1.1 什么是企业级应用

企业级应用的含义并非是为某一个企业所开发的应用程序，而是指为政府、组织、大型企业或机构创建和开发的规模比较大的、运行于局域网或广域网的应用程序。通常，企业级应用具备以下一些特点。

(1) 分布式。因为使用用户数量和方式的缘故，企业级应用程序往往是运行在网络环境下的，或者运行于局域网，或者运行于广域的 Internet，把分布于一个范围内，乃至世界各地的用户连接到一起。

(2) 快速反应。当今社会飞速发展，能够反映社会状况的数据和运行规则在不断地发生着变化。企业级应用的数据和功能要能够根据社会的变化，随时发生更新和改动。

(3) 安全性。软件系统对于用户和企业而言越来越重要，它的正常运行对于政府、组织、企业或机构及所有用户而言，其重要性不言而喻，所以其运行代码和运行所使用的数据的安全性必须是有保障的。

(4) 可扩展性。为了适应日益发展的管理、商务等社会需求，以及用户数量的不断增加，企业级应用软件系统的性能和数据存储能力，以及软件的功能必须是可扩展的。

1.1.2 当今的计算机应用环境

经历了 70 多年的发展历程，计算机系统已经成为社会生活中不可或缺的重要技术和支撑平台。计算机处理器的运算速度越来越快，处理能力越来越强，软件的种类越来越多，应用软件的规模越来越大。当今的计算机技术与计算机应用呈现以下几个明显的特点。

(1) 硬件系统平台多种多样。从日常生活中经常见到的 PC，到为某些专门领域和应用而设计的工作站、小型机，为了大规模的数据处理和数值计算而研制的大型超级计算机，再到以苹果公司的 iPhone、iPad 等为代表的移动设备，各种计算机硬件产品琳琅满目。这

些产品的体系结构不同，指令集不同，使用目的不同，生产厂商不同，互相之间几乎都不兼容。

(2) 系统软件平台“群雄割据”。PC、工作站、小型机、大型超级计算机和移动设备的操作系统不同，数据库服务器不同，编程语言不同，开发工具不同，互相之间几乎也都不兼容。

(3) 网络协议和网络体系结构尚未统一。尽管到目前为止，TCP/IP 占据主导地位，但是距离社会实际需求的统一和集成的网络系统的目标依然相差甚远。

(4) 当今所处的时代是一个信息化的时代，信息化已经进入了 Internet 时代，计算机技术和各种信息化技术已经深入社会生活的各个领域，信息技术几乎无处不在。

1.1.3 Internet 时代计算机应用的主要矛盾

Internet 时代，WWW 的发展和需求增长异常迅猛，软件的运行平台要求高度统一到“分布式”“异构”的 Internet 平台上来，高度统一的 Internet 平台与多种多样的软件、硬件平台的矛盾是一个重要问题。已有软件的集成和新软件的开发成为一个发展瓶颈，“软件危机”的说法早在 20 世纪 60 年代就已经被人们提出来了。采用怎样的方式才能将已有的这些系统集成起来，并且能够保持良好的运行是一个十分现实而又十分困难的事情。解决这种现实问题的措施必须是为 Internet 平台和 WWW 应用服务的，必须是在现有的硬件条件下实现的，必须是在现有的操作系统条件下实现的，必须充分考虑到以往已有的数量庞大的软件产品的再利用问题，必须考虑到今后进行软件开发设计过程中的成本问题和开发便捷性问题。

1.1.4 中间件的概念

鉴于这种日益增长的软件需求和 Internet 时代应用系统开发所面对的实际问题，计算机界内的有识之士提出了中间件的概念。中间件的概念是自“软件危机”以来，继“面向对象的程序设计”概念提出之后，软件开发手段的又一次变革。

什么是中间件？中间件（Middleware）是基础软件的一大类，属于可复用软件的范畴。“中间”指的是其处于操作系统软件、网络、数据库之上，应用软件之下，总的作用是为处于其上层的应用软件提供运行与开发的环境。通过中间件，应用程序可以工作于多种硬件平台和操作系统环境。

一个普遍被接受的定义是 IDC（International Data Corporation，国际数据公司）给出的：“中间件是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源，中间件位于客户机/服务器的操作系统之上，管理计算资源和网络通信。”这个对于中间件概念的阐述明确指出了中间件是一类软件而不是某一种软件，其作用是在系统软件和应用软件之间实现连接，实现通过不同的接口共享资源。

中间件具有以下几个特征：(1) 独立于系统，满足大量应用的需要；(2) 用于分布式环境，支持分布式计算；(3) 运行于多种硬件和操作系统平台，具有网络通信功能，提供网络、硬件、操作系统的透明性的交互功能，可以实现应用之间的互操作；(4) 支持标准的协议，支持标准的接口；(5) 本身是开发平台，可以在其上进一步开发应用程序系统。

中间件的优越性主要表现在以下一些地方：(1) 在应用开发方面，通常可以节省 25%~

60%的应用开发费用，如果配合使用商用构件，最多可节省 80%的开发费用；(2) 在系统运行过程中，可节省 50%的初期资金和运行费用；(3) 开发周期，使用标准的商业中间件可缩短开发周期 50%~75%；(4) 在项目开发上可以有效减少项目开发风险，失败率低；(5) 合理运用资金，利用中间件可以将原有的系统“改头换面”，增加功能模块，成为 Internet/Intranet 系统，有效地保护已有的软件资源；(6) 应用集成，标准化的中间件可以集成现有的应用、新的应用和新购买的商用构件；(7) 系统维护，中间件的开发代价高，但购买商业中间件只需付出产品价格的 15%~25%的维护费，从而降低维护费用，具体费用要看供应商的价格和购买数量；(8) 质量，标准中间件在接口方面应该是清晰和规范的，能够有效地保证应用系统的质量；(9) 技术革新，标准的商业中间件厂商应责无旁贷地把握技术方向和技术革新，因此在软件的革新和升级方面，中间件可以做得更好；(10) 增加产品吸引力，不同的商业中间件提供不同的功能模块，合理使用，可使应用软件“流光溢彩”；(11) 优化软件开发，开放的中间件标准可以让更多的厂商和个人中间件开发者加入，有利于软件开发的优化。

采用中间件技术可以促进软件标准化，简化最终开发，保护已有投资，稳定应用环境，集成和协调应用软件。执行中间件的一个关键途径是信息传递。

对于中间件的类型划分尚无一个明确的标准，通常按照中间件的作用，大致可以将中间件分为两大类：把支持单个的应用系统或解决一类问题的中间件称为底层中间件，一般包括交易中间件、应用服务器、消息中间件、数据访问中间件；把用于与各种应用系统关联、完成系统整合的中间件称为高层中间件，一般包括企业应用集成中间件、工作流中间件、门户中间件等。

目前，中间件技术已成为大数据、云计算等应用的必不可少的工具。

1.1.5 中间件的发展

中间件的迅猛发展是最近几年的事情，但是与很多人的想象不同，中间件的理论概念出现于 20 世纪 70 年代，产品也很早就有了。一般将诞生于 1983 年、由 AT&T 公司的 Bell 实验室开发的 Tuxedo 系统作为中间件的诞生标志。Tuxedo 解决了分布式交易事务控制问题，开始成为网络应用的基础设施，这是最早的交易中间件。Tuxedo 系统被 Novell 公司从 AT&T 公司随着 UNIX 系统一起买走，后来它又被卖给了 BEA 公司，随着 BEA 公司被 Oracle 公司收购，Tuxedo 系统现在已经归于 Oracle 公司旗下。

1993 年，IBM 公司发布了消息队列服务 MQ 系列产品，解决了分布式系统异步、可靠、传输的通信服务问题，消息中间件正式诞生。这个产品也是比较有代表性的中间件产品。

1995 年，Java 语言面世，它提供了跨平台的通用的网络应用服务，成为今天中间件的核心技术。Java 是第一个天生的网络应用平台，特别是 J2EE 发布以来，Java 从一个编程语言演变为网络应用架构，成为应用服务平台的事实标准和应用服务器中间件，成为中间件技术的集大成者，也成为事实上的中间件的核心。

2002 年，Microsoft 公司发布 .NET，加入中间件的市场竞争。由于 .NET 还不是一个完全开放的技术体系，所以其竞争力还有待市场的进一步确认。

围绕着中间件，Apache 组织、IBM 公司、Oracle (BEA) 公司、Microsoft 公司各自发展了较为完整的软件产品体系。中间件技术创建在对应用软件部分常用功能的抽象上，将常

用且重要的过程调用、分布式组件、消息队列、事务、安全、连接器、商业流程、网络并发、HTTP 服务器、Web 服务等功能集于一身，或者在不同品牌的不同产品中分别完成。一般认为在商业中间件及信息化市场上主要存在 Java 阵营、Microsoft 阵营、开源阵营。阵营的区分主要体现在对下层操作系统的选择及对上层组件标准的制定上。目前主流商业操作系统主要来自 UNIX、苹果公司和 Linux 的系统及 Microsoft 公司的 Windows 系列。Java 阵营的主要技术提供商来自 IBM 公司、Sun 公司（已被 Oracle 收购）、Oracle 公司、BEA 公司（已被 Oracle 收购）及其合作伙伴。Microsoft 阵营的主要技术提供商来自 Microsoft 公司及其商业伙伴，开源阵营则主要来自如 Apache、SourceForge 等组织的共享代码。

2000 年前后，中国国内的软件行业也看到了中间件技术产品巨大的市场价值，已有公司和机构加入了中间件技术的行列中。其中比较有代表性的包括深圳金蝶（Kingdee）国际软件集团（香港联交所主板上市公司，股票代码：0268）、山东浪潮电子信息产业股份有限公司（深圳证券交易所上市公司，股票代码：000977）等。

1.2 Web 应用的发展

1.2.1 静态网页技术

Web 应用是 Internet 技术出现之后发展的一种新的应用技术。最初的 Web 应用只是一个一个的站点（Website），其中包含很多网页。用户在使用这些站点时无须安装任何专用程序，可以直接使用网络浏览器来访问网络服务器，打开网络服务器上面存储的页面，实现信息浏览。这些页面就是静态网页。

静态网页一般都没有后台数据库，没有可执行代码，没有交互内容，其内容通常都是固定的，任何用户登录服务器看到的页面内容都是一样的。绝大部分静态网页都是采用 HTML 格式的文本文件写成的，或者是采用与 HTML 文本兼容的格式写成的，如 XHTML、XML 等。

HTML（HyperText Markup Language，超文本标记语言）是由 HTML 命令组成的描述文本，可以用来说明网页页面所包含的文字、图形、动画、声音、表格、超链接等被网络浏览器解析再现的内容。HTML 技术出现之后，其技术内容迅速增加，所包含的命令内容迅速扩张，形成了一整套技术规范和技术标准。

JavaScript 是一种基于对象和事件驱动并具有相对安全性的客户端脚本语言，同时也是一种广泛用于客户端 Web 开发的脚本语言，常用来给 HTML 网页添加动态功能，如响应用户的各种操作。完整的 JavaScript 实现包含三部分：ECMAScript、文档对象模型、字节顺序记号。JavaScript 就是为适应动态网页制作的需要而诞生的一种新的编程语言，JavaScript 的出现使得网页页面能够具有交互性。在 HTML 基础上，使用 JavaScript 可以开发交互式 Web 网页。JavaScript 的出现使得网页和用户之间实现了一种实时性的、动态的、交互性的关系，使网页包含更多活跃的元素和更加精彩的内容。

还有一种叫作 CSS（Cascading Style Sheets，层叠样式表）的技术可以让网站管理员为 HTML 文档或 XML 应用等结构化文档添加字体、间距和颜色等样式的标记性语言，这也是在网页维护和使用中经常用到的。

1.2.2 动态网页技术

静态网页由于其先天不足，大大限制了 Web 应用的使用和发展，Internet 技术的普及和发展使得越来越多的应用程序有必要转到 Web 上面去，于是克服了静态网页技术不足动态网页技术就应运而生了。最早出现的动态网页技术是 CGI，而热门的动态网页开发技术是 ASP、JSP、PHP 三种。

CGI (Common Gateway Interface, 通用网关接口) 是一段部署和运行在服务器上的程序，提供同客户端 HTML 页面的接口。绝大多数的 CGI 程序被用来解释、处理来自表单的输入信息，并且在服务器产生相应的处理，或者将相应的信息反馈给浏览器。CGI 程序使网页具有交互功能。CGI 带来的好处是弥补了 HTML 的不足，提供许多 HTML 无法做到的功能，让 Web 页面浏览者与服务器进行交互。但由于 CGI 应用程序运行在浏览器可以请求的服务器系统上，执行时需要占用服务器 CPU 的运算时间和内存，如果有成千上万的这种程序同时运行，会对服务器系统的运算性能提出极高的要求，服务器系统存在崩溃的风险。

ASP、JSP、PHP 三种动态网页开发技术则是在 CGI 的基础上出现的，这三种技术都改进了 CGI 的先天不足，成为流行至今的动态网页开发技术。ASP、JSP、PHP 三种动态网页开发技术都可以很容易地实现对数据库服务器的连接和访问。

ASP (Active Server Pages, 动态服务器页面) 是 Microsoft 公司开发的代替 CGI 脚本程序的一种应用，它可以与数据库和其他程序进行交互，是一种简单、方便的编程工具。ASP 的网页文件的格式是“.asp”，是一种服务器端脚本编写环境，可以用来创建和运行动态网页或 Web 应用程序。ASP 网页可以包含 HTML 标记、普通文本、脚本命令及 COM (Component Object Model, 组件对象模型) 组件等。利用 ASP 可以向网页中添加在线表单等交互式内容，也可以创建使用 HTML 网页作为用户界面的 Web 应用程序。利用 ASP 可以突破静态网页的一些功能限制，实现动态网页技术，同时由于 ASP 文件是包含在 HTML 代码所组成的文件中的，易于修改和测试。服务器上的 ASP 解释程序会在服务器端执行 ASP 程序，并将结果以 HTML 格式传送到客户端浏览器上，因此使用各种浏览器都可以正常浏览 ASP 生成的网页。ASP 的一个缺陷是只能在 Microsoft 的操作系统环境下运行。ASP 技术的后继者 ASP.NET 已经于 2002 年面世。

JSP (JavaServer Pages, Java 服务器页面) 是由 Sun 公司及其合作伙伴共同参与建立的一种动态网页技术标准。JSP 技术有些类似于 ASP 技术，它是在传统的网页 HTML 文件中插入 Java 程序段和 JSP 标记，从而形成 JSP 网页文件，格式是“.jsp”。JSP 也具有与 ASP 相似的优点，也可以生成使用各种浏览器都可以正常浏览的网页。用 JSP 开发的 Web 应用是跨平台的，能在多种操作系统上运行。与传统的 CGI 方式相比，JSP 的后台实现逻辑是基于 Java 组件的，并且将应用逻辑与页面表现分离，使得应用逻辑能够最大程度得到复用，从而进一步提高了开发效率。由于 JSP 的后台是完全基于 Java 技术的，所以其安全性由 Java 的安全机制予以保障。

PHP (Personal Home Page) 这个名称已经正式更名为 Hypertext Preprocessor，即超文本预处理器，由 Rasmus Lerdorf 创建于 1994 年，语言的风格有些类似于 C 语言，被广泛地运

用在开发各种中小型网站上。PHP 独特的语法混合了 C、Java、Perl 及 PHP 自创的新的语法，它可以比 CGI 或 Perl 更快速地执行动态网页。PHP 将程序嵌入 HTML 文档中去执行，用 PHP 做出的动态页面执行效率比完全生成 HTML 标记的 CGI 要高许多；PHP 还可以执行编译后代码，编译可以加密和优化代码运行，使代码运行更快。PHP 具有非常强大的功能，所有 CGI 的功能 PHP 都能实现，而且 PHP 支持几乎所有流行的数据库及操作系统。最重要的是 PHP 可以用 C、C++ 进行程序的扩展。由于 PHP 具有开放的源代码，本身又是免费的，相对于其他语言来说，编辑简单，实用性强，程序开发快，运行快，技术本身学习快，所以更适合初学者。由于在运行过程中 PHP 消耗的系统资源相当少，因此近年来被广泛地运用在规模不大、功能不强、访问量不太大的小型网站上，很多使用其他技术开发的小型网站都改用 PHP 技术，这使得 PHP 的市场占有率在最近几年有所上升。

1.2.3 C/S 体系结构模式与 B/S 体系结构模式

在计算机网络应用程序连接模式中，早期比较常用的为 C/S (Client/Server, 客户机/服务器) 体系结构模式，把运行客户程序的机器称为“客户机”(Client)，把运行服务器程序的机器称为“服务器”(Server)。一般也将 C/S 体系结构模式称为二层结构。C/S 体系结构模式可以充分利用两端硬件环境的优势，将任务合理分配到 Client 端和 Server 端来实现，降低了系统的通信开销。C/S 结构的优点是能充分发挥客户端 PC 的处理能力，很多工作可以在客户端处理后再提交给服务器，客户端的响应速度快。但其在运行使用过程中也有明显的劣势，就是较高的运行成本和客户端软件升级成本。尤其是在计算机应用进入 Internet 时代之后，面临着 Web 应用这种新的方式，升级维护的成本劣势便日益显现。

为了适应 Web 应用的客观要求，一种新的体系结构模式应运而生，这就是目前被广泛采用的 B/S (Browser/Server, 浏览器/服务器) 体系结构模式。这是随着 Internet 技术的兴起，对 C/S 体系结构模式的一种变化和改进。在这种体系结构模式下，用户工作界面是通过 WWW 浏览器来实现的，网络浏览器作为系统的客户端，极少部分事务逻辑在前端即浏览器端实现，但主要事务逻辑在服务器端实现，在系统安全性方面获得了大幅度的提高。B/S 体系结构模式最大的优点就是可以在任何地方进行操作而不用安装任何专门的软件。当系统需要进行升级和维护时，仅需对服务器端进行升级维护，而不必像 C/S 体系结构模式那样需要更新大量的客户端。B/S 体系结构模式的另一个优点是允许选择不同的服务器操作系统，不管选用哪种操作系统都可以让大部分使用 Windows 作为桌面操作系统的计算机用户不受影响，这就使得最流行的免费 Linux 操作系统快速发展起来。现在的趋势是使用 B/S 体系结构模式的应用管理软件，只需安装在 Linux 服务器上即可，而且安全性也很高。B/S 体系结构模式的一个明显缺点是由于应用程序的主要事务逻辑都在服务器端实现，浏览器端仅完成一些界面逻辑，从而造成应用服务器运行数据负荷较重，当系统一旦发生服务器“崩溃”等问题时，后果不堪设想。因此，许多应用程序系统都备有数据库存储服务器，以防止应用服务器发生意外时数据遭受破坏。

目前的分布式应用程序系统大多采用 B/S 体系结构模式。

1.2.4 多层应用体系结构

随着近几年来 Web 应用需求的日益增强,以及中间件技术的发展,特别是 Java 技术体系的发展,开始出现了三层乃至多层体系结构的应用程序系统,并且呈现大规模流行的趋势。三层和多层体系结构是在 B/S 体系结构模式的基础上发展起来的,在 B/S 体系结构模式的浏览器端和服务器端的中间,增加了一个或多个用于进行事务处理、监测、信息处理和过滤、Web 服务等业务工作的工作层,从而形成了一个新的工作体系结构。

在多层应用体系结构中,层(Tier)是一个划分不同的业务分工的概念,各个层之间分工明确,根据企业信息系统各个组成部分在功能上的区别,将整个应用系统划分为表示层、业务逻辑层和数据层,其中的业务逻辑层又可根据不同的企业需求,进一步划分为流程层、服务层、逻辑层、操作层、映射层等多个层,划分方法也不尽相同。通常将这些不同的功能层划分统称为多层应用体系结构。

表示层也称客户层或客户端,用来提供呈现在客户端的人机交互界面,以及完成用户信息的输入工作和用户需求结果的呈现工作,表示层可以像一般的 Web 服务一样构造页面,实现页面之间的链接、导航等。在一般情况下,多层应用体系结构的表示层都是用 Web 浏览器呈现的,无须安装专门的客户端,所以被形象地称为“瘦客户”。

业务逻辑层是多层应用体系结构的核心部分,承担了应用程序系统的主要工作。业务逻辑层的工作包括:(1)处理应用程序与具体业务内容相关的逻辑计算,如银行、保险、财务等方面的计算工作,这是业务逻辑的核心任务;(2)数据库的访问和数据提取工作,实现用户与数据库之间对话的桥梁功能;(3)用户所提交数据的解析、映射工作;(4)用户所需要的信息的界面组织构造工作;(5)分布式系统的管理工作,实现系统的负载均衡、安全隔离等功能。业务逻辑层可以根据企业应用系统的具体需求,进行功能的细分,划分成多个功能层,形成多层应用体系。

数据层提供数据的存储服务,一般就是数据库管理系统。由于在业务逻辑层中可以实现灵活的数据映射,所以多层应用体系结构的数据层可以选择使用多种数据库平台,包括市场上流行的 Oracle、SQL Server、MySQL、DB2 等,还可以支持不同的数据库模型,如关系数据库、对象数据库,以及基于 XML 的层次数据库等。

多层应用体系结构的优势主要包括:(1)安全,业务逻辑层隔离了用户和数据库,有效地保护了数据,防止了对数据库的各种侵害;(2)稳定,业务逻辑层缓冲了用户对数据库的实际连接,加之采用了如数据连接池等技术,使得系统的工作稳定性大大增强;(3)易维护,当业务规则发生变化,或者软件的处理模块需要升级时,仅需修改业务逻辑层的相关模块,其他模块、客户端和数据库基本不用改动;(4)响应速度快,由于负载均衡和业务逻辑层对数据的缓存能力,系统整体上提高了对用户提交的请求的响应速度;(5)灵活的扩展方式,当业务扩大、用户的数量增加时,可以在中间层部署更多的应用服务器,提高对客户端的响应和有效连接数。

多层应用体系结构的开发技术规范目前主要有两个:COM+和 CORBA(Common Object Request Broker Architecture,公共对象请求代理架构)。COM+是面向 Windows 平台的,而 CORBA 则提供跨平台的能力。随着近年来软件和硬件技术的不断更新,特别是计算机领域反垄断的呼声日益高涨,开源组织不断涌现,跨硬件平台、跨网络环境、跨操作系统及跨数