高等教育规划教材

Java EE开发技术与实践教程

聂艳明　刘全中　李宏利　邹青　编著

提供电子教案

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 开发技术与实践教程

聂艳明　刘全中　李宏利　邹　青　编著

机械工业出版社

本书共分为五大部分，涵盖了 Java EE 的主流开发技术。第一部分介绍了 Java Web 开发模型及其演化、Java EE 基础服务。第二部分对 Java Web 基础开发技术，如 JSP、Servlet、EL、JSTL 以及 MVC 进行了阐述。第三部分集中论述基于轻量级 SSH（Struts2+Spring+Hibernate）框架开发的原理和技术，特别是三者之间的整合方法。第四部分着重探讨了基于经典 Java EE 框架（JSF+EJB+JPA）开发的原理和方法，重点在于其架构性理念和规范。最后一部分给出了针对同一项目的三种不同开发技术方案，以期读者能获得 Java EE 应用分层开发技术的整体性理解。本书的每个章节都配有拓展参考阅读文献，以指导读者进一步深入学习。

本书既可作为高等院校计算机软件开发技术课程的教材，也可作为软件开发从业人员的技术参考书。

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：2966938356，电话：010-88379739）。

图书在版编目（CIP）数据

Java EE 开发技术与实践教程 / 聂艳明等编著. —北京：机械工业出版社，2014.9
高等教育规划教材
ISBN 978-7-111-48043-3

Ⅰ. ①J… Ⅱ. ①聂… 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（2014）第 219125 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　师沫迪　　责任校对：张艳霞
责任印制：刘　岚

涿州市京南印刷厂印刷

2015 年 1 月第 1 版 • 第 1 次印刷
184mm×260mm • 23.5 印张 • 580 千字
0001－3000 册
标准书号：ISBN 978-7-111-48043-3
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
服务咨询热线：（010）88379833
读者购书热线：（010）88379649
封面无防伪标均为盗版

网络服务
机工官网：www.cmpbook.com
机工官博：weibo.com/cmp1952
教育服务网：www.cmpedu.com
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。

希望教育界的专家和老师能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 已经成为企业级应用开发的首选技术解决方案之一。本书以 MyEclipse 8.5、MySQL 5.6、Tomcat 6.0 或 JBoss 6.0（用于 EJB 项目）作为开发和运行测试环境，以对比的叙述方式介绍了主流的 Java EE 开发技术及其实践。

本书从 Java Web 开发模型及其演化、Java EE 多层架构及其基础服务的简单介绍入手，深入浅出地叙述了当前流行的 Java EE 开发技术的原理、方法及实践：1）基于 MVC 的 Java Web 开发基础模型（JSP+Servlet+JavaBean）；2）基于 SSH（Struts2+Spring+Hibernate）轻量级框架；3）基于 JSF+EJB+JPA 的经典 Java EE 框架。在本书的最后一个部分，针对同一项目采用上述三种 Java Web 开发技术方案分别予以实现，以期读者对于 Java EE 分层架构及各层对应模型或框架的技术选型有一个整体性的把握。

本书体现了面向实用型软件工程人才培养的课程改革的方向，在关注 Java Web 开发基础的前提下，紧跟业界需求，重点介绍基于轻量级 SSH 框架的开发技术，同时从应用架构和规范标准的角度兼顾了经典 Java EE 框架。对于本书的使用：1）可以任选其中一种或两种 Java Web 开发模型或框架作为主要内容，而将其他部分作为辅助参考，这种使用方式建议授课 30 课时、实验 10 课时；2）可以包含所有内容，建议授课 40 课时、实验 20 课时，同时规划课余时间用于课程项目开发实践，以切实强化学生对基础概念和开发实践的掌握。

本书中包含的所有示例代码都是在 MyEclipse 8.5、MySQL 5.6 以及 Tomcat 6.0 或 JBoss 6.0 环境下调试运行通过的。第 14 章给出一个完整的示例，以帮助读者顺利地完成开发任务。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内容实施。作为教材，每章后附有小结和拓展阅读书目建议，以指导读者对相关主题的深入学习。

本书的第 1、2、9、13 和 14 章由聂艳明编写；第 8 章和第 11 章由刘全中编写；第 3～7 章由邹青编写；第 10 章和第 12 章由李宏利编写。全书由聂艳明统稿，张阳教授审定。在本书的编写过程中，徐雷等同学承担了书稿的文字校对工作。本书的顺利出版，还要感谢西北农林科技大学信息工程学院的李书琴院长等领导以及其他老师所给予的大力支持和帮助。李建良、张宏鸣、赵建邦、王美丽、英明、王湘桃、毛锐、颜永丰等软件工程系的同仁，也为本书的编写提供了宝贵建议。

需要本书第 14 章示例项目代码以及全书示例的读者，请到出版社网站上下载。

本书出版得到了陕西省教育科学“十二五”规划 2014 年度课题“面向实用型软件工程人才培养的 Java EE 课程教学与实践改革探索研究”（课题编号：SGH140541）的资助与支持。

由于时间仓促、作者水平有限，书中难免存在不妥之处，恳请读者批评指正并提出宝贵意见。作者 Email：yanmingnie@nwsuaf.edu.cn。

编　者

目　录

第三部分　轻量级框架 SSH

第四部分 经典 Java EE 框架

第五部分 案例项目开发实践

第一部分　Java EE 基础及服务

第 1 章　Java Web 开发模型及其演化

随着计算机软硬件技术的发展以及应用需求的变迁，应用模式也随之发生演化，例如从单机、C/S、B/S 再到云应用。当前 B/S 应用（亦称为 Web 应用）为主流，基于 Java 的 Web 开发模型也经历了 3 个主要阶段，即原始阶段、模型阶段和框架阶段，多层架构的 Java EE（包括表现层、业务逻辑层、持久层和数据库层）可以被看做是 Java Web 应用开发的框架规范。

1.1　应用模式演化

1.1.1　单机应用

单机应用（Standalone Application）为全部应用任务独立运行于一台机器上的应用，如记事本、画图以及 MS Word 等。基于数据库的应用系统中，应用任务一般可以分为三类：用户界面表示、业务逻辑处理以及应用数据存取。在单机应用中，上述三类任务都被部署并运行于同一台机器上。如基于 MS Access 的简单数据库应用也属于单机应用。

在计算机网络还不是很普及的年代，应用大都为单机版。随着计算机网络的发展和业务需求自身的改变，应用程序逐步转向多个系统协同的网络版，如随后将会讨论到的 C/S 应用、B/S 应用、云应用等。

1.1.2　C/S 应用

C/S（Client/Server，客户机/服务器）架构为分布式的软件体系结构的一种。C/S 架构的基本原则是将计算机应用任务分解成多个子任务，由多台计算机分工完成，即采用“功能分布”的原则。除了完成自身的任务外还需请求诸如网络连接、数据存取、消息收发等服务的程序称为客户机程序（Client），提供相应服务的程序则称为服务器程序（Server），客户机和服务器通过网络（可以是 Intranet 或 Internet）连接，C/S 架构示意见图 1-1。通过 C/S 架构，可以在尽量降低系统通信开销的同时，充分利用两端软硬件环境的优势，将应用任务合理分配到客户端和服务器来执行。特别的，具有分布要求的业务应用需要通过网络连接的多台机器协作完成。

根据客户机和服务器任务的分配重点，C/S 架构中的客户机又可分为胖客户机（Rich Client）和瘦客户机（Thin Client）。如客户机只是负责用户界面表示（其他两类任务由服务器负责），则为瘦客户机；相应的，如客户机负责用户界面和业务逻辑处理（服务器只负责应用数据存取），则称为胖客户机。

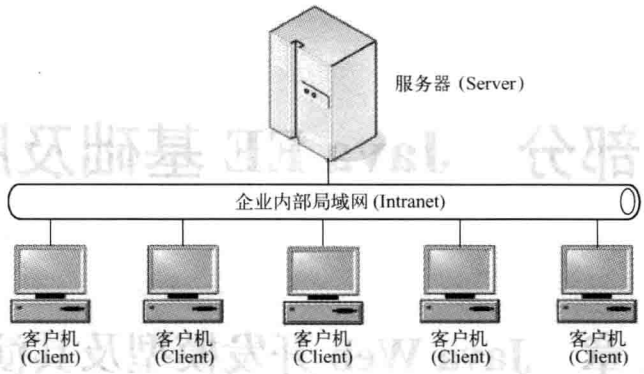

图 1-1　C/S 架构

C/S 应用就是基于 C/S 架构的应用，客户程序与服务器之间的交互协同一般体现为协议通信，如基于 FTP 的文件传输系统、基于 HTTP 的 WWW 系统（其中，如微软 IE 等浏览器为客户机）、基于 POP3 和 SMTP 的邮件系统等都属于 C/S 应用。早期基于 PowerBuilder、Dephi、VC、VB 等开发的数据库应用系统也都属于 C/S 应用。

与单机应用相比，C/S 应用的主要优势是实现了业务逻辑的分布式处理。另外，C/S 应用的交互性好，基于 C/S 架构实现的业务分布协作具有良好的可伸缩性（Scalability）。同时，C/S 应用的缺点也是显而易见的。首先，系统（特别是客户程序）的安装、调试、升级和维护存在较大困难。其次，由于业务分布于多个客户程序而不是由服务器集中统一控制，存在一定的安全隐患。当然，通过特定的客户机与服务器间的通信协议和加密认证措施，可以减小甚至是提高系统的安全性，如银行业务系统。

1.1.3　B/S 应用

B/S（Browser/Server，浏览器/服务器）架构是 Web 兴起之后的一种软件体系结构，本质上是一种瘦客户端的 C/S 架构。其工作原理与 C/S 架构类似，也实现了应用业务分布式部署和运行，并通过由网络连接的多台机器交互协作完成。只是 B/S 架构模式将客户端统一为 Web 浏览器，如微软的 IE、Google 的 Chrome、苹果的 Safari、Mozilla Firefox 以及 Opera 等。B/S 架构示意见图 1-2。

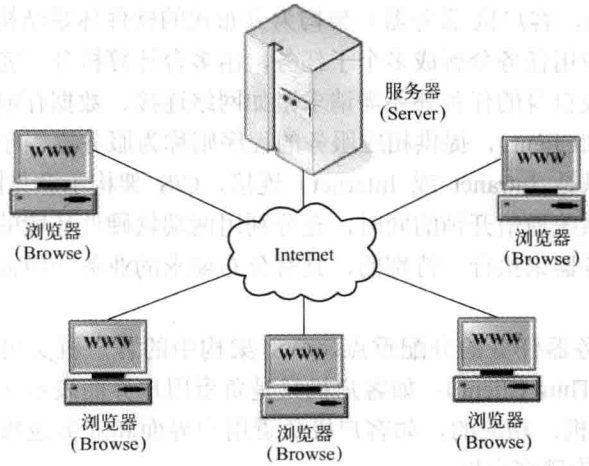

图 1-2　B/S 架构

B/S 应用就是基于 B/S 架构的应用，客户程序与服务器之间的通信协议为 HTTP。B/S 应用中的业务集中于服务器，浏览器主要负责系统的 UI 渲染和交互。在基于数据库的应用中，服务器需要负责 UI 的服务器端生成、业务逻辑处理以及应用数据存取，相应可以划分为 Web 服务器、应用服务器和数据库服务器，分别部署和运行对应类型的业务，如图 1-3 所示。可以根据业务复杂程度、业务量大小以及具体需求选择服务器端的架构。

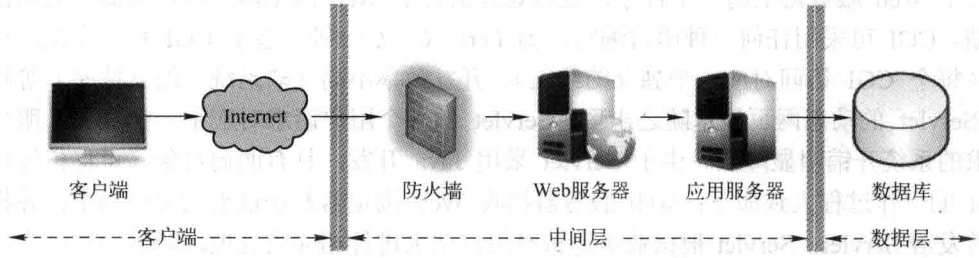

图 1-3　B/S 架构中的服务器

如同 C/S 应用，B/S 应用也可以实现业务逻辑的分布式处理。而且，B/S 应用具有业务扩展简单方便、开发效率高、升级维护简单方便等特点。由于 B/S 应用的 UI 表现是基于 HTML 页面的，所以传统模式下的特殊功能、个性化要求（特别是 UI 方面）都难以实现。另外，B/S 应用在速度和安全性上也存在一定的问题。

1.1.4　云应用

按照美国国家标准与技术研究院（NIST）的定义，云计算（Cloud Computing）是一种按需付费的服务模式，该模式提供可用、便捷、按需的网络访问，进入可配置的计算资源共享池（资源包括网络、服务器、存储、应用软件、服务），这些资源能够被快速提供，只需投入很少的管理工作或与服务提供商进行很少的交互。云计算通常涉及通过互联网来提供动态易扩展、虚拟化的资源（如图 1-4 所示），是分布式计算、并行计算、效用计算、网络存储、虚拟化、负载均衡等传统计算机和网络技术发展融合的产物，是继 20 世纪 80 年代大型计算机到 C/S 大转变之后计算模式的又一次巨变。

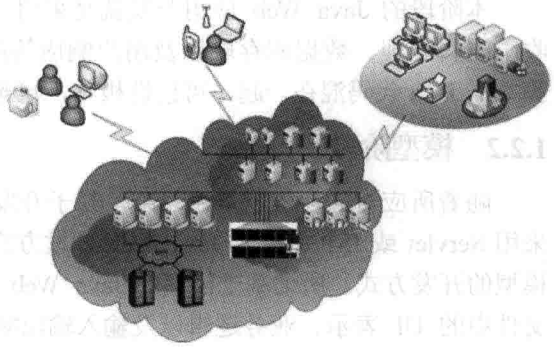

图 1-4　云应用环境示意图

云应用（Cloud Application）可看做是基于云计算概念和架构构建的应用，是云计算技术在应用层的体现。云应用与云计算的最大区别在于，云计算作为一种宏观技术发展概念而存在，而云应用则是直接面对客户解决实际问题的产品，如云存储、云地图导航、云杀毒、云渲染以及搜索引擎等。云应用的工作原理是把传统软件“本地安装、本地运算”的使用方式变为“即取即用”的服务，通过互联网或局域网连接并操控远程服务器集群，完成业务逻辑或运算任务。云应用的主要载体为互联网，以瘦客户端或智能客户端的形式展现。云应用可以帮助用户降低 IT 成本，还能提高工作效率，因此传统软件向云应用转型的发展革新浪潮已是不可阻挡。

1.2 Java Web 开发模型演化

1.2.1 原始阶段

早期的动态网页主要采用公共网关接口（Common Gateway Interface，CGI）技术。CGI 为运行于 Web 服务器上的一个程序，负责处理来自于 Web 浏览器的用户所提交的数据并提供反馈。CGI 可采用任何一种语言编写，如 Perl、C、C++等。鉴于 CGI 程序存在运行性能不佳（每个 CGI 访问对应一个独立的线程）、开发效率不高（缺乏统一的支持库）等缺点，基于 Servlet 的动态网页技术随之出现。Servlet 的各个用户请求对应于一个线程，服务器处理请求的系统开销明显降低。由于 Servlet 采用 Java 开发，具有面向对象、可移植等特点。Servlet 的工作过程大致如下：Web 服务器接收 Web 浏览器发送过来的用户请求，并将请求数据转发给 Servlet；Servlet 根据业务逻辑对用户请求进行相应的处理，随后动态产生响应反馈给客户端浏览器。这样就可以完成系统和用户的一次交互。

Servlet 并未彻底克服 CGI 程序编写复杂（相比于 HTML 脚本编写）的缺陷，这样就出现了 Java 服务器端网页（Java Server Pages，JSP）技术，JSP 可以看做是嵌入了 Java 代码的 HTML 页面。JSP 的工作原理与 Servlet 本质上相同，只是降低了动态网页开发的技术门槛，使得 HTML 页面设计人员经过初步培训就可以完成基本动态网页的编写。与 JSP 不同，动态服务器页面（Active Server Page，ASP）基于微软技术平台，无法移植到非 Windows 环境。另一种动态网页技术超文本预处理器（Hypertext Preprocessor，PHP）与 Linux、Apache 和 MySQL 的集成解决方案 LAMP（Linux + Apache + MySQL + PHP）是目前较为流行的 Web 应用开发框架。

本阶段的 Java Web 应用开发就是采用 Servlet 或 JSP，将处理用户请求的请求数据接收、逻辑处理、数据库存取以及用户响应等所有的代码置于一个文件中，程序中的 HTML 脚本与 Java 代码混在一起，可读性极差，也难以维护。

1.2.2 模型阶段

随着所应对的业务越来越复杂，易于开发和维护成为应用系统的首要技术指标。显然，采用 Servlet 或 JSP 的原始 Java Web 开发方式就显得很不适应现实需求了，进而演变为基于模型的开发方式。所谓基于模型的 Java Web 开发方式，就是将原处于同一个 Servlet 或 JSP 文件中的 UI 表示、业务逻辑以及输入输出响应分割为不同类型的组件，并通过这些组件的协同实现应用业务。同时，关注 Java Web 应用系统的开发便利性和可维护性。按照分割的方式和粒度，Java Web 开发模型可以划分为以下两种。

1．模型 1：JSP/Servlet + JavaBean

该模型将 Java Web 应用中业务逻辑处理、数据库存取等部分的代码抽取为 JavaBean，JSP 或 Servlet 则主要负责用户请求接收、页面跳转控制、用户响应生成以及输入/输出表示。工作流程相应为 JSP/Servlet 接收用户请求，然后调用 JavaBean 进行业务处理和数据存取，最后生成响应返回给客户端。

2．模型 2：JSP + Servlet + JavaBean

模型 2 对模型 1 继续进行改进，在将 Java Web 应用中业务逻辑处理、数据库存取等部

分的代码抽取为 JavaBean 的同时，对模型 1 中的 JSP 或 Servlet 的代码做进一步分割，即输入/输出 UI 表示部分的代码纳入 JSP 组件，用户请求接收、页面跳转控制及用户响应生成部分的代码则纳入到 Servlet 组件中。工作流程相应为 Servlet 接收用户请求，然后调用 JavaBean 进行业务处理和数据存取，最后生成响应并以 JSP 的形式返回给 Web 浏览器或控制到其他 Servlet 的跳转。

无论是模型 1 还是模型 2，通过代码分割都显著地提高了 Java Web 应用系统的开发效率和可维护性。但是，由于很多支撑性技术代码如数据库存取、页面流程控制、输入/输出的预处理、操作日志维护等都需要开发人员自行编写，开发效率、可复用程度以及代码质量都有待进一步提高。对于大型的、较为复杂的企业级 Java Web 应用，尤为如此。

1.2.3 框架阶段

框架（Framework）是整个或部分系统的可重用设计，表现为一组抽象构件及构件实例间交互的方法。框架是可被应用开发者定制的应用骨架。前者是从应用方面，而后者是从目的方面给出的定义。应用框架并不包含构件应用程序本身，而是实现某应用领域通用完备功能（相对于应用所独有的业务功能等）的底层服务及各部分组件交互规范。应用框架强调的是软件的设计重用性和系统的可扩充性，以缩短大型应用软件系统的开发周期，提高开发质量为目标。应用框架可以从两个方面来理解：首先，框架是一种规范，规定了所包括组件构成及其之间的交互规范；其次，框架又是一种基础服务的实现，为应用开发人员提供相应 API 及配置进行基于框架的应用系统开发。

针对 Java Web 应用开发，目前存在很多框架。包括应对 UI 交互、页面流程跳转等的框架，如 Struts、JSF（JavaServer Faces）等；应对业务逻辑处理的框架，如 Spring、EJB 等；应对数据持久的框架如 Hibernate、MyBatis、JPA（Java Persistence API，Java 持久 API）等。在实际 Java Web 应用开发过程中，会根据项目的自身特点和具体需求选择合适框架进行组合。目前主流的 Java Web 应用开发框架组合可分为两类：轻量级（Struts+Spring+Hibernate，SSH）和经典（JSF+EJB+JPA）。其中，以 Spring 为核心的轻量级框架 SSH 在开发效率方面存在较大优势，而以 EJB 为核心的经典框架则主要适用于对分布和可伸缩性（Scalability）要求较高的企业级 Java Web 应用（经典 Java EE 应用）开发。

> 经典意义上的 Java EE 是上文提到的以 EJB 为核心，由 Sun（现已被 Oracle 收购）提出的。“Java EE”中的第一个 E 即为企业（Enterprise），与“EJB”中的 E 含义相同。除了针对业务组件分布部署的 EJB 外，还包括分别针对 UI 表现和数据持久的 JSF 和 JPA（将在后续的章节中进行详细说明）。为简便起见，通常将基于其他框架，如 SSH 等的 Java Web 应用开发也称为 Java EE 开发。

1.3 Java EE 多层架构

1.3.1 概述

Java EE 采用多层架构，以应对应用系统业务和技术的复杂性。Java EE 多层架构模型将 Java EE 服务划分成多层（Tier）：表现层（Presentation Tier）、业务层（Business Tier）和持久层（Persistence Tier）。Java EE 多层架构以及各层技术框架如图 1-5 所示。其中，表现层（客户端）和数据层是作为 Java EE 应用整体结构的参与层，并非 Java EE 架构的重点。以下

只对 Java EE 的表现层（服务器端）、业务层和持久层分别做一个简单介绍。

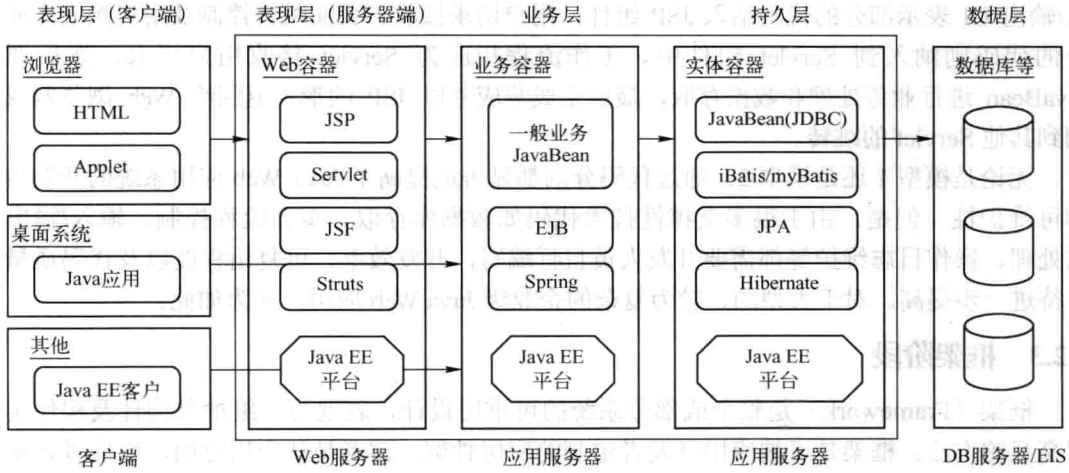

图 1-5　Java EE 分层架构以及各层技术框架

1.3.2　表现层

作为 Java EE 架构的第一层，表现层的主要功能就是实现用户交互、界面表示以及页面流程控制等。表现层收集客户请求数据，调用处于第二层即业务层中的核心服务，并生成客户请求响应。表现层的主要组件（JSP 和 Servlet）需要使用 Web 容器所提供的服务，并由 Web 容器管理。JSP/Servlet 以及辅助 JavaBean 可作为最基础的表现层技术方案，常见的表现层框架有 Tapestry、Struts、Spring MVC、JSF 等。

1.3.3　业务层

业务逻辑层为 Java EE 架构的中间层，主要实现核心业务逻辑服务。该层由大量服务组件（Components）组成，亦称为组件层。业务逻辑层接收来自于表现层的服务请求，向持久层请求数据存取服务，并将处理结果返回给处于表现层的请求者。作为服务（Service）组件的 JavaBean 可作为最基础的业务逻辑层的技术方案，常见的业务逻辑层框架有 Spring、EJB 等。基于框架的业务层组件需要使用相应的业务组件容器（如 IoC、AOP 等）所提供的服务，并由容器来管理业务对象的生命周期。

1.3.4　持久层

持久层构成 Java EE 架构的第三层，主要负责数据库等应用数据的存取。持久层为业务逻辑层提供对业务数据的存取服务。包括数据存取对象（Data Access Object，DAO）和持久对象（Persistence Object，PO）的 JavaBean 可作为最基础的持久层的技术方案，常见的持久层框架有 Hibernate、MyBatis、TopLink、JPA 等。基于框架的持久层组件需要使用相应的实体容器（如 Hibernate、JPA 等容器）所提供的 ORM、持久、事务等服务，并由实体容器管理实体对象的生命周期。

1.4　本章小结

本章内容对应用模式和 Java Web 开发模式的演化以及 Java EE 多层架构进行了简要介

绍。随着计算机技术的发展以及需求的变迁，应用模式由早期的单机模式过渡到 C/S 和 B/S 架构直至当前的云应用。Java Web 应用的开发模式也经历了从基于 JSP/Servlet 的原始阶段到基于业务分割的模型阶段和框架阶段的发展历程。为了应对应用系统业务和技术的复杂性，Java EE 采用多层架构（即表现层、业务层、持久层和数据层），并为每层架构制定了组件和交互规范。可以选择合适的框架实现组合来开发 Java EE 应用。

关于 Java EE 多层架构可以进一步参考 Oracle 公司网站的 Java EE 专题。对于各层框架实现的技术细节可以参考相应的网站。

拓展阅读参考：

[1] Oracle Corporation. Java EE[OL]. http://www.oracle.com/technetwork/java/javaee/overview/index.html.

[2] The Apache Software Foundation. Struts[OL]. http://struts.apache.org.

[3] GoPivotal, Inc. Spring[OL]. http://projects.spring.io/spring-framework/.

[4] Red Hat, Inc. Hibernate[OL]. http://hibernate.org/.