DVD

DVD语音视频讲解光盘
50个开发实例及程序源代码
60个语音视频讲解实现及环境搭建过程
附赠7种网络和应用工具软件

Java EE 5

完全学习手册

黄开枝 许勇 王黎 等编著

本书内容特色

涵盖从事Java EE开发所要掌握的知识
应用JSP、JSF、Servlet、RMI、EJB和Swing
介绍EJB组成、会话bean、实体和消息驱动bean
结合JSP、Servlet、Swing和EJB技术开发网络购书系统
窗内网（www.itzcn.com）提供技术支持

清华大学出版社

DVD

DVD语音视频讲解光盘

50个开发实例及程序源代码

60个语音视频讲解实现及环境搭建过程

附赠7种网络和应用工具软件

Java EE 5

完全学习手册

黄开枝 许勇 王黎 等编著

本书内容特色

涵盖从事Java EE开发所要掌握的知识

应用JSP、JSF、Servlet、RMI、EJB和Swing

介绍EJB组成、会话bean、实体和消息驱动bean

结合JSP、Servlet、Swing和EJB技术开发网络购书系统

窗内网（www.itzcn.com）提供技术支持

清华大学出版社

北　京

内 容 简 介

本书将带领读者进入 Java EE 平台开发的世界，由浅入深地学习各项知识。全书共分为 4 篇 13 章，内容依次为 Java EE 概述、搭建 Java EE 环境、Servlet 技术、JSP 技术、JDBC 技术、JSF 客户端技术、JNDI 和 RMI 开发、EJB 技术架构、会话 Bean、实体、JMS 和消息驱动 Bean、JavaMail 技术和 Java EE 实例网上购书系统。配书光盘提供了全书实例完整源代码和软件配置等重要操作的视频文件。

本书适合于中、高级 Java EE 5 开发人员，特别适合于有编程基础，希望全面学习 Java EE 5 技术，提高实际应用能力的读者群体。

图书在版编目（CIP）数据

Java EE 5 完全学习手册 / 黄开枝等编著. —北京：清华大学出版社，2009.4
ISBN 978-7-302-19428-6

Ⅰ.J…　Ⅱ. 黄…　Ⅲ. JAVA 语言－程序设计　Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2009）第 015564 号

责任编辑：夏兆彦
责任校对：徐俊伟
责任印制：王秀菊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京鑫海金澳胶印有限公司
装 订 者：三河市李旗庄少明装订厂
经　　销：全国新华书店
开　　本：190×260　**印　张**：30.75　**字　数**：762 千字
附光盘 1 张
版　　次：2009 年 4 月第 1 版　　**印　　次**：2009 年 4 月第1 次印刷
印　　数：1～4000
定　　价：59.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103　　产品编号：029958-01

前　言

企业版 Java 平台 Java EE 5（Java Platform, Enterprise Edition）的出现使得 Java 企业应用的开发变得简单和快捷。Java EE 5 平台的目的在于为开发者提供一系列强大的 API，减少开发时间，降低应用复杂性并且提高应用的性能。Java EE 5 是 J2EE 的新生，是 J2EE 开发进一步简化的升级版本。

Java EE 5 既保持 J2EE 强大的功能，又简化了开发任务。Java EE 5 平台引入了一个简化的编程模型。在 Java EE 5 技术中，XML 部署描述符是可选择的。相反，开发者可以将信息作为注解（Annotation）直接输入到 Java 源文件中。Java EE 服务器在部署和运行时会对组件进行配置。这些注解通常被嵌入到由部署描述符提供的程序数据中。使用注解时，规范信息被直接放置到代码中与它所影响的程序元素相邻的位置。

1. 本书内容介绍

本书将带领读者进入 Java EE 平台开发的世界，由浅入深地学习各项知识。全书共分为 4 篇。

第 1 篇为概述篇，包括第 1 章和第 2 章。介绍了 Java EE 的体系结构和搭建 Java EE 环境的过程，并以实例的方式演示 Java EE 程序的运行，是初学者必读的部分。

第 2 篇为基本技术篇，包括 3、4、5、6 和 7 章。该篇重点介绍 Java EE 中所必需的各项基本技术，如 JSP、JSF、Servlet 和 RMI 等。

第 3 篇为 EJB 篇，包括 8、9、10 和 11 章。重点介绍 EJB 组成、会话 Bean、实体和消息驱动 Bean。本篇以较大的篇幅总结了 EJB 3.0 的使用，是本书学习的重点。

第 4 篇为实践篇，包括 12 和 13 章。介绍使用 JSP、Servlet、Swing 和 EJB 等技术开发分布式应用程序的过程。在实现过程中采用记事本作为开发工具，最大程度地反映了 Java EE 实现的方方面面，并严格遵循软件的开发流程。

2. 本书主要特色

本书全面介绍 Java EE 5 开发和应用相关的知识，具有下面的特点。

- **内容全面**　本书是一本大全性质的 Java EE 5 编程图书，突出介绍 Java EE 5 的开发知识。读者学习本书之后，可以全面掌握 Java EE 5 的开发实践知识。
- **实例丰富**　全书每个知识点的讲解都配有大量可实际运行的实例，读者可以边学习边实践，快速、全面地掌握 Java EE 5 的开发方法和技巧。书中最后一篇还提供了典型开发案例，覆盖了 Java EE 5 开发和部署网站的全部过程。

本书配套光盘提供了本书源代码（包括 4 个完整的应用程序）和 Java EE 5 软件配置和使用的教学视频。读者只要按照书中的范例上机练习，举一反三，就可以根据自己的需要开发出各种功能强大的应用。

3. 本书读者对象

本书结构清晰，语言通俗易懂。本书适合于中、高级 Java EE 5 开发人员，特别适合于有编程基础，希望全面学习 Java EE 5 技术，提高实际应用能力的读者群体。本书也可作为高等院校的教学用书和相关培训机构的培训教材。

除了封面署名人员之外，参与本书编写的还有于永军、张秋香、李乃文、张仕禹、夏小军、赵振江、李振山、李文才、吴越胜、李海庆、何永国、李海峰、陶丽、吴俊海、安征、张巍屹、崔群法、王咏梅、康显丽、辛爱军、牛小平、贾栓稳、王立新、苏静、赵元庆、郭磊、徐铭、李大庆、王蕾、张勇、郝安林、郭新志、牛丽平、唐守国等。

在编写过程中难免会有疏漏，欢迎读者与我们联系，帮助改正提高。

目　录

第1篇　概述篇

第2篇　基本技术篇

第 4 篇 实践篇

第1篇 概 述 篇

第1章 Java EE概述

内容摘要 | Abstract

J2EE/Java EE（Java 2 Platform Enterprise Edition）是建立在J2SE（Java 2 Platform Standard Edition）的基础上，为企业级应用提供了完整、稳定、安全和快速的Java平台。Java EE提供的Web开发技术主要支持两类软件的开发和应用。一类是做高级信息系统框架的Web应用服务器（Web Application Server），另一类是在Web应用服务器上运行的Web应用（Web Application）。

本章首先分析Java EE体系结构，然后在此基础上介绍容器、组件等与Java Web开发技术密切相关的基本概念。最后阐述组成Java EE体系结构的各层所采用的Java Web开发技术以及所提供的各种服务等。

学习目标 | Objective

- 了解Web技术发展历史
- 理解Java EE产生的必然性
- 掌握Java EE的规范和概念
- 了解Java EE的特点
- 掌握单层、两层和多层系统框架
- 熟练掌握Java EE技术框架
- 掌握Java EE容器
- 掌握Java EE常用组件技术

1.1 Java EE发展简介

Java EE框架的产生是Web技术发展到一定阶段的产物。对于Java EE框架，可以从多个角度介绍，如计算机开源技术的发展、计算机框架技术的发展等。为了更好地理解Java EE技术，本节从Web技术的产生和发展开始介绍，一直介绍到Java EE的产生。

1.1.1 Web技术发展历程

众所周知，Web这个Internet上最热门的应用架构是由Tim Berners-Lee发明的。Web的前

身是 1980 年 Tim Berners-Lee 负责的 Enquire（Enquire Within Upon Everything）项目。1990 年 11 月，第一个 Web 服务器 nxoc01.cern.ch 开始运行，Tim Berners-Lee 在自己编写的图形化 Web 浏览器“World Wide Web”上看到了最早的 Web 页面。1991 年，CERN（European Particle Physics Laboratory）正式发布了 Web 技术标准。目前，与 Web 相关的各种技术标准都由著名的 W3C（World Wide Web Consortium）组织管理和维护。

从技术层面看，Web 架构的精华有 3 处：用超文本技术（HTML）实现信息与信息的连接；用统一资源定位技术（URI）实现全球信息的精确定位；用新的应用层协议（HTTP）实现分布式的信息共享。这 3 个特点无一不与信息的分发、获取和利用有关。其实，Tim Berners-Lee 早就明确无误地告诉大家：“Web 是一个抽象的（假想的）信息空间”。换言之，作为 Internet 上的一种应用架构，Web 的首要任务就是向人们提供信息和信息服务。

然而，在 Web 应用日新月异的今天，许多 Web 开发人员似乎已经忘记了 Web 架构的设计初衷。开发人员在自己开发的网站或 Web 应用中大肆堆砌各种所谓的“先进”技术，但是最终用户能够在这些网站或应用中获得的有价值的信息却寥寥无几。这个问题绝不像评论者常说的“有路无车”或“信息匮乏”那么简单。从信息服务角度来说，评价一种 Web 开发技术优劣的标准只有一个，就是看这种技术能否在最恰当的时间和最恰当的地点，以最恰当的方式，为最需要信息的人提供最恰当的信息服务。

Web 架构是一种典型的分布式应用架构。Web 应用中每一次信息交换都要涉及到客户端和服务端两个层面。因此，Web 开发技术也常被分为客户端技术和服务端技术两大类。首先发来介绍一下客户端技术和服务器端技术的发展过程。

1. 客户端技术

Web 客户端的主要任务是展现信息内容。HTML 语言是信息展现的最有效载体之一。作为一种实用的超文本语言，HTML 最早可追溯到 20 世纪 40 年代。最初的 HTML 语言只能在浏览器中展现静态的文本或图像信息。这满足不了人们对信息丰富性和多样性的强烈需求。最终的结果是，静态技术向动态技术的转变成为 Web 客户端技术发展的必然趋势。

Web 出现后，GIF 第一次为 HTML 页面引入了动态元素。但更大的变革来源于 1995 年 Java 语言的问世。Java 语言天生就具备的平台无关的特点，让人们找到了在浏览器中开发动态应用的捷径。1996 年，著名的 Netscape 浏览器在其 2.0 版中增加了对 JavaApplets 和 JavaScript 的支持。Microsoft 的 IE 3.0 也在这一年开始支持 Java 技术。喜欢动画、喜欢交互操作、喜欢客户端应用的开发人员可以用 Java 或 JavaScript 语言随心所欲地丰富 HTML 页面的功能。为了让纯 Microsoft 技术能与 JavaScript 抗衡，Microsoft 还为 1996 年的 IE 3.0 设计了另一种后来也声名显赫的脚本语言——VBScript 语言。

1996 年底，W3C 提出了 CSS 的建议标准。同年，IE 3.0 引入了对 CSS 的支持。CSS 大大提高了开发者对信息展现格式的控制能力。1997 年的 Netscape 4.0 不但支持 CSS，而且增加了许多 Netscape 公司自定义的动态 HTML 标记。1997 年，Microsoft 发布了 IE 4.0，并将动态 HTML 标记、CSS 和动态对象模型（DHTML Object Model）发展成一套完整、实用、高效的客户端开发技术体系。Microsoft 称其为 DHTML。同样是实现 HTML 页面的动态效果，DHTML 技术无需启动 Java 虚拟机或其他脚本环境，就可以在浏览器的支持下获得更好的展现效果和更高的执行效率。

为了在 HTML 页面中实现音频、视频等更为复杂的多媒体应用，1996 年的 Netscape 2.0

成功地引入了对 QuickTime 插件的支持，插件这种开发方式也迅速风靡于浏览器的世界。在 Windows 平台上，Microsoft 将客户端应用集成的赌注押到了 20 世纪 90 年代中期问世的 COM 和 ActiveX 身上。1996 年，IE 3.0 正式支持在 HTML 页面中插入 ActiveX 控件的功能，为其他厂商扩展 Web 客户端的信息展现方式开辟了一条自由之路。1999 年，Realplayer 插件先后在 Netscape 和 IE 浏览器中取得了成功。与此同时，Microsoft 自己的媒体播放插件 Media Player 也被预装到各种 Windows 版本中。同样值得纪念的还有 Flash 插件的横空出世。20 世纪 90 年代初期，Jonathan Gay 在 FutureWave 公司开发了一种名为 Future Splash Animator 的二维矢量动画展示工具。1996 年，Macromedia 公司收购了 FutureWave，并将 Jonathan Gay 的发明改名为 Flash。从此，Flash 动画成为 Web 开发者表现自我、展示个性的最佳方式。

2. 服务器端技术

与客户端技术从静态向动态的发展过程类似，Web 服务端的开发技术也由静态向动态逐渐发展、完善起来。最早的 Web 服务器简单地响应浏览器发来的 HTTP 请求，并将存储在服务器上的 HTML 文件返回给浏览器。一种名为 SSI（Server Side Includes）的技术可以让 Web 服务器在返回 HTML 文件前，更新 HTML 文件的某些内容，但其功能非常有限。第一种真正使服务器根据运行时的具体情况动态生成 HTML 页面的技术是 CGI（Common Gateway Interface）技术。1993 年，CGI 1.0 的标准草案由 NCSA（National Center for Supercomputing Applications）提出。1995 年，NCSA 开始制定 CGI 1.1 标准。1997 年，CGI 1.2 也被纳入议事日程。CGI 技术允许服务端应用程序根据客户端的请求动态生成 HTML 页面，使客户端和服务端的动态信息交换成为可能。随着 CGI 技术的普及，聊天室、论坛、电子商务、信息查询、全文检索等的 Web 应用蓬勃兴起，人们终于可以享受到信息检索、信息交换、信息处理等更为便捷的信息服务了。

早期 CGI 程序大多是编译后的可执行程序，其编程语言可以是 C、C++、Pascal 等任何一种通用的程序设计语言。为了简化 CGI 程序的修改、编译和发布过程，人们开始探寻用脚本语言实现 CGI 应用。Perl 结合了 C 语言的高效以及 sh、awk 等脚本语言的便捷的特点，适用于 CGI 程序的编写。1995 年，第一个用 Perl 写成的 CGI 程序问世。很快，Perl 在 CGI 编程领域的风头就盖过了 C 语言。随后，Python 等著名的脚本语言也陆续加入了 CGI 编程语言的行列。

1994 年，Rasmus Lerdorf 发明了专用于 Web 服务端编程的 PHP（Personal Home Page Tools）语言。与以往的 CGI 程序不同，PHP 语言将 HTML 代码和 PHP 指令合成为完整的服务端动态页面。Web 应用的开发者可以用一种更加简便、快捷的方式实现动态 Web 功能。1996 年，Microsoft 借鉴 PHP 的思想，在其 Web 服务器 IIS 3.0 中引入了 ASP 技术。借助 Microsoft Visual Studio 等开发工具在市场上的成功，ASP 迅速成为了 Windows 系统下 Web 服务端的主流开发技术。当然，以 Sun 公司为首的 Java 阵营也不甘示弱。1997 年，Servlet 技术问世。1998 年，JSP 技术诞生。Servlet 和 JSP 的组合（还可以加上 JavaBean 技术）让 Java 开发者同时拥有了类似 CGI 程序的集中处理功能和类似 PHP 的 HTML 嵌入功能。此外，Java 的运行时编译技术也大大提高了 Servlet 和 JSP 的执行效率。这也正是 Servlet 和 JSP 被后来的 J2EE 平台吸纳为核心技术的原因之一。

2000 年以后，随着 Web 应用的日益复杂，人们逐渐意识到，单纯依靠某种技术无法达到快速开发、快速验证和快速部署的最佳境界。研究者开始尝试将已有的 Web 开发技术综合起来，

形成完整的开发框架或应用模型，并以此来满足各种复杂的应用需求。

1.1.2 Java EE 框架产生

Web 服务端开发技术的完善使开发复杂的 Web 应用成为可能。在此起彼伏的电子商务大潮中，为了适应企业级应用开发的需求，同时也是为了给最终用户提供更可靠、更完善的信息服务，两个最重要的企业级开发平台——J2EE 和.NET 平台在 2000 年前后分别诞生于 Java 和 Windows 阵营中，并随即在企业级 Web 开发领域展开了你死我活的拼争。平台之争让整个 Web 世界在最近的几年里不得安宁。但从某种意义上说，也正是这种针锋相对的竞争关系促使 Web 开发技术以前所未有的速度向前发展。

1. J2EE 框架产生

J2EE 是纯粹基于 Java 的解决方案。1998 年，Sun 发布了 EJB 1.0 标准。EJB 为企业级应用中的数据封装、事务处理、交易控制等功能提供良好的技术基础。至此，J2EE 平台的三大核心技术 Servlet、JSP 和 EJB 都已先后问世。1999 年，Sun 正式发布了 J2EE 的第一个版本。紧接着，遵循 J2EE 标准，为企业级应用提供支撑平台的各类应用服务软件相继涌现出来。IBM 的 WebSphere、BEA 的 WebLogic 都是这一领域里最为成功的商业软件平台。随着开源运动的兴起，JBoss 等开源世界里的应用服务新秀也吸引了许多用户的注意力。到 2003 年，Sun 的 J2EE 版本已经升级到 1.4 版，其中 3 个关键组件的版本也演进到了 Servlet 2.4、JSP 2.0 和 EJB 2.1。至此，J2EE 体系及相关的软件产品已经成为 Web 服务端开发的一个强有力的支撑环境。

2. Java EE 框架产生

Java 已经被应用到了企业、桌面、Web、移动等各个领域，其中覆盖面最广的 J2EE 及相关产品被广泛应用到企业中。但从 1999 年诞生的第一个 J2EE 版本一直到 J2EE 1.4 版本，总被人们不断地抱怨。这并不是因为它不够强大。恰恰相反，正是因为它太强大了，强大得让人难以使用。实现一个简单的 J2EE 程序，需要大量的配置文件，尽管有些配置文件不是必需的。

Sun 公司一直在试图改变这一切，但一直未能如愿。2002 年，J2EE 1.4 推出后，J2EE 的复杂程度达到顶点。尤其是 EJB 2.0，开发和调试的难度非常大。也许是要下决心改变这一切，或者是受到市场和开发人员的压力，Sun 终于在 2006 年 5 月份正式发布了 J2EE 1.5（现改名为 Java EE 5）规范，并宣称 Java EE 5 将是 Java EE 史上最简单的版本，将大大降低开发难度。

现有的 Java EE 技术非常成熟。实际上，Java EE 5 的主基调“简化开发”也暗示这一点。可以看出，Java EE 5 在试图简化开发者视图（客户视图）。即让开发者能够更方便、高效地使用 Java EE 技术，而不是引入新的，Java EE 容器级的功能。

但是，Java EE 中的功能高度集成，无法单独使用其中的一部分。许多 Servlet、Java 数据库和 Java Server Pages 开发人员一般只使用 Java EE 的某些相关特性。但 Java EE 规范要求必须使用所有特性。为了使程序可以正常运行，开发人员不得不建立一个复杂的工程来满足这些要求。这是因为 Java EE 仍然保持了 20 世纪 90 年代后期的编程方法，Java EE 仍以 API 为中心。

1.2 Java EE 概述

Java EE（Java 2 Platform Enterprise Edition）是 Java 平台企业版，是一套技术架构。Java EE 可提高应用程序的可移植性、安全与再用价值，其核心是一组技术规范与指南。Java EE 的产生使开发人员只需要注重商业逻辑与架构设计。

1.2.1 Java EE 简介

Java EE 是 J2EE 版本的后续版本，是 J2EE 技术的新生和发展。Java EE 技术具有 J2SE 平台的所有功能，同时还提供对 EJB、Servlet、JSP、XML 等技术的全面支持。Java EE 的最终目标是成为一个支持企业级应用开发的体系结构，简化企业解决方案的开发、部署和管理等复杂问题。事实上，Java EE 已经成为企业级开发的工业标准和首选平台。Java EE 是一个标准而不是一个产品，各个平台开发商按照 Java EE 规范开发不同的 Java EE 应用服务器。推出 Java EE 框架的目的是为了克服传统 C/S 模式的弊端，迎合 B/S 架构的潮流。

Java EE 不是 Java 2 标准版（J2SE）的替代品。J2SE 是 Java EE 的核心部分，它为 Java EE 提供了基本的语言框架。正如后面将会看到的那样，Java EE 由很多层组成。而 J2SE 只是 Java EE 所有组件的基础。作为 Java 开发人员，可能已经学会使用 Swing 或者 AWT（Abstract Window Toolkit）的组件来建立用户界面。对于 Java EE 应用程序，仍然可以用这些组件来建立用户界面。此外还可以开发基于 HTML 的用户界面。正因为 J2SE 是 Java EE 的核心，所以过去所学的任何有关 Java 的知识仍然都相当有用。

此外，Java EE 还有另一套创建用户界面的 API，称为 JSF（Java Server Faces）。这是 Java 最新的技术之一。Java EE 平台也支持开发应用程序的中间层部分，包括业务逻辑和与后端数据源的连接。可以利用熟悉的 J2SE 组件和 API，结合 Java EE 的组件和 API 来构建这部分应用。

为了开发企业级的应用，Java EE 定义了大量的服务，用于实现企业级别的应用和大量的基础结构。编写具有可扩展性的、健壮的、安全的和易于维护的分布式应用程序，需要用到大量的系统级功能。这些重要的基础结构模块包括了安全性、数据库访问和事务控制等系统级功能。其中，安全性保证了用户只可以访问应用中那些被赋予相应权限的部分。数据库访问是最基本的组件之一，便于应用程序存储和读取数据。对事务的支持则保证在恰当的时间更新相应的数据。

毫无疑问，Java EE 解决了这个问题。它定义了一套容器、连接器和组件。Java EE 不仅弥补了兼容性的问题，而且还是基于公开的规范。所以为 Java EE 平台编写的应用可以运行在任何 Java EE 兼容的实现之上。Sun 公司的 Java EE 软件开发包（Software Development Kit，SDK）提供了一个参考实现，其中包括一个工作模型。因为这是 Sun 公司完全按照规范实现的，并且是免费的，所以本书也采用了它。Java EE 框架所支持的服务以及工作流程，如图 1-1 所示。

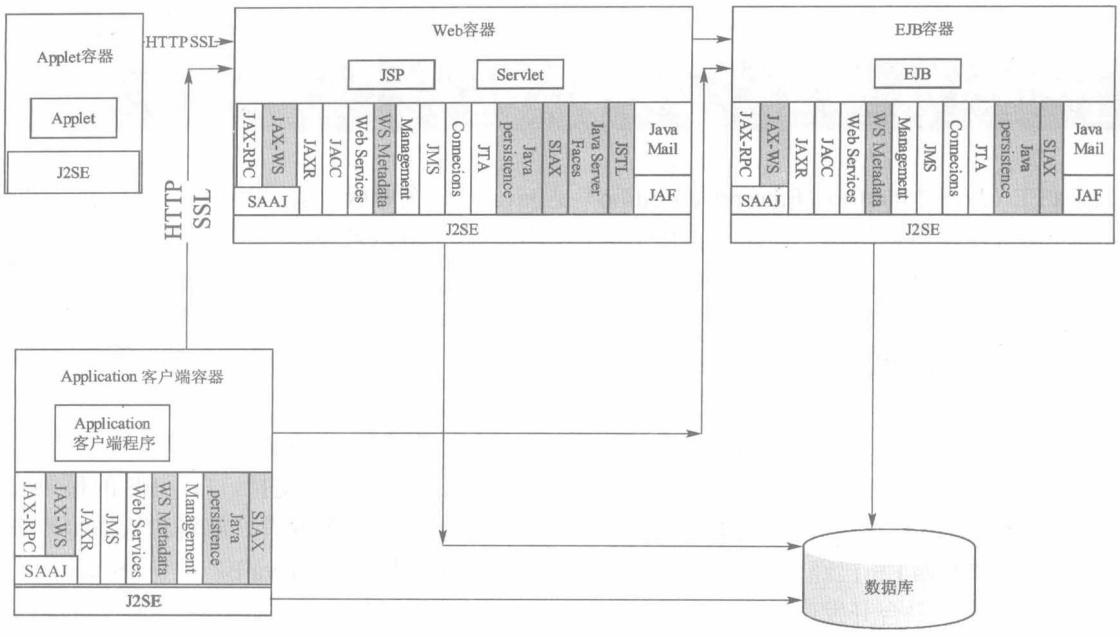

图 1-1　Java EE 5 框架图

图中灰色加黑部分是 Java EE 5 新的功能。可以看到，在 Web 层主要加入了 JSF 这个新的表现层框架。同时还引入了新的 Java 持久化标准。这个标准正在由 EJB 3.0 专家组制定。

值得指出的是，这个 Java 持久化标准也可以嵌入在 Web 层调用。所以，它肯定不会从属于 EJB 标准。这样，当前所有的 Java 持久层标准，如 JDBC/JDO/Hibernate/Entity Bean 将可能统一，从而减少用户的架构选择烦恼。

从上图可以看出，无论 Web 结构或 EJB 结构，提供实现的功能相差不多。这样业务核心组件就可以根据需要部署在 Web 或 EJB 容器中运行，而不依赖具体的 Java EE 容器了。

1.2.2　Java EE 5 规范新概念

对于熟悉 Java EE 系列框架的读者来说，无论 J2EE 1.2 还是 J2EE 1.4 版本，甚至是现在的 Java EE 5 版本，其核心规范都是 EJB（Enterprise Java Beans）。EJB 包含 3 种 Bean，分别是会话 Bean（Session Bean）、实体（J2EE 以前版本称为实体 Bean，Java EE 5 之后称为实体）和消息驱动 Bean（Message Driven Bean），前面两个比较重要。

Java EE 5 相比较于其以前版本，即 J2EE 1.4 和 J2EE 1.2 等，其规范中添加了如下新的概念。

1．标注取代部署描述符

Java EE 5 平台不需要任何部署描述符（Servlet 规范所需的部署描述符 web.xml 文件除外），从而简化了部署过程。其他部署描述符，如 ejb-jar.xml 以及在 web.xml 中与 Web 服务相关的条目，在 Java EE 5 中可以不使用。众所周知，J2EE 1.4 部署描述符通常很复杂，在编写这些部署描述符时很容易出错，特别是对于初学者。Java EE 5 采用“标注”来代替部署描述符，大大

简化了 Java EE 程序的开发。

标注（也称为注解，注释）是 Java 修饰符，与代码中指定的 public 和 private 类似。例如，EJB 3 规范（Java EE 5 规范的子集）为 Bean 类型、接口类型、资源引用、事务属性、安全性等定义了标注。JAX-WS 2.0 规范为 Web 服务提供了一组类似的标注。有些标注用来生成工件，另外一些标注用来描述代码，还有一些标注用来提供增强服务，如安全性或特定于运行时的逻辑。总之，Java EE 5 平台为以下任务（以及其他任务）提供标注。

- 定义和使用 Web 服务。
- 开发 EJB 软件应用程序。
- 将 Java 技术类映射到 XML。
- 将 Java 技术类映射到数据库。
- 将方法映射到操作。
- 指定外部依赖关系。
- 指定部署信息，其中包括安全属性。

标注使用“@”字符来标记。当创建了使用 Java EE 5 中的标注的类型时，将在生成的代码中提供相关的占位符。例如，创建无态会话 Bean 时，需要生成以下代码，其中包括@Stateless()标注。

```
package mypackage;
import javax.ejb.*;
@Stateless()
public class HelloWorldSessionBean implements mypackage.HelloWorld
SessionLocal {
}
```

2. 简化的 EJB 软件开发

使用新的 EJB 3.0 API 可以减少开发者的工作量，从而使开发者可以更轻松地进行软件开发。换句话说，就是使用了更少的类和代码。这是因为容器承担了更多的工作。下面是新 EJB 3.0 API 的一些功能和优点。

- **只需很少的类和接口** 不再需要 EJB 组件的 Home 接口和对象接口，容器负责公开必要的方法。只需提供业务接口。可以使用标注来声明 EJB 组件，并且通过容器来管理事务。
- **不再需要部署描述符** 可以在类中直接使用标注，为容器提供以前在部署描述符中定义的依赖关系和配置信息。如无任何特殊说明，容器将使用默认规则来处理最常见的情况。
- **查找简单** 可以通过 EJBContext 直接在类中查找 JNDI 名称空间中的对象。
- **简化了对象关系映射** 新的 Java 持久性 API 允许使用 POJO 中的标注将 Java 对象映射到关系数据库，从而使对象关系映射变得更简单透明。

在可视化开发工具 IDE 中，可以对 Enterprise Beans 进行编码，就像对其他 Java 类进行编码一样。方法是：使用代码编辑器提示实现正确的方法，并使类与其接口保持同步。不必使用特殊命令和对话框生成诸如业务方法或 Web 服务操作之类的内容。虽然这些命令仍可以帮助大