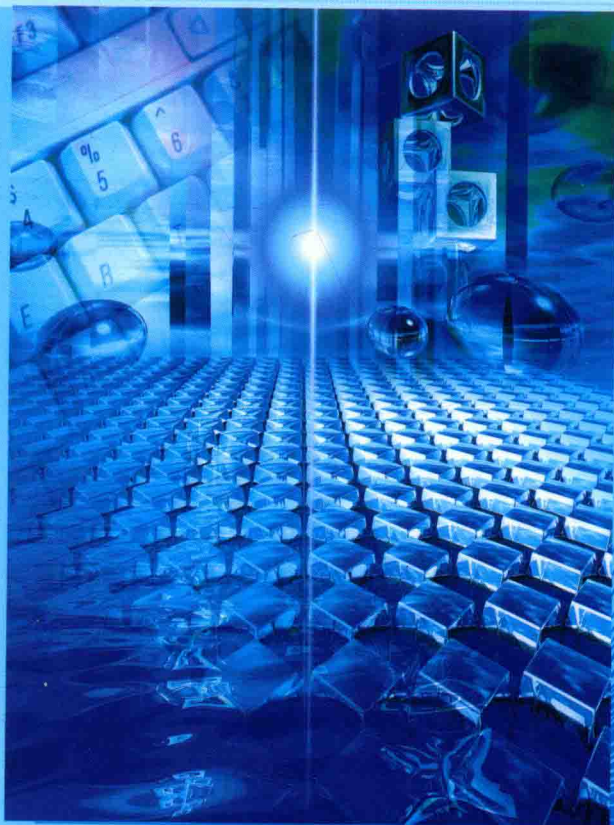




“十三五”江苏省高等学校重点教材
应用型本科院校计算机类专业校企合作实训教材系列

Java EE 框架技术与案例教程

杨种学 王小正 主编



南京大学出版社



“十三五”江苏省高等学校重点教材（本书编号：2016-1-056）
应用型本科院校计算机类专业校企合作实训教材系列

Java EE 框架技术与案例教程

主 编 杨种学 王小正
副主编 侯 青 徐家喜 朱 杰



【微信扫码】
本书导学，领你入门

南京大学出版社

图书在版编目(CIP)数据

Java EE 框架技术与案例教程 / 杨种学, 王小正主编.

— 南京: 南京大学出版社, 2017. 12

应用型本科院校计算机类专业校企合作实训教材系列

ISBN 978-7-305-19722-2

I. ①J… II. ①杨… ②王… 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(2017)第 316646 号

出版发行 南京大学出版社

社 址 南京市汉口路 22 号 邮编 210093

出版人 金鑫荣

丛 书 名 应用型本科院校计算机类专业校企合作实训教材系列

书 名 **Java EE 框架技术与案例教程**

主 编 杨种学 王小正

责任编辑 王秉华 王南雁 编辑热线 025-83597482

照 排 南京理工大学资产经营有限公司

印 刷 南京人民印刷厂有限责任公司

开 本 787×1092 1/16 印张 17.75 字数 430 千

版 次 2017 年 12 月第 1 版 2017 年 12 月第 1 次印刷

ISBN 978-7-305-19722-2

定 价 44.80 元

网 址: <http://www.njupco.com>

官方微博: <http://weibo.com/njupco>

官方微信号: njupress

销售咨询热线: (025)83594756

* 版权所有, 侵权必究

* 凡购买南大版图书, 如有印装质量问题, 请与所购
图书销售部门联系调换

在当前的信息时代和知识经济时代,计算机科学与技术的应用已经渗透到国民生活的方方面面,成为推动社会进步和经济发展的重要引擎。

随着产业进步、学科发展和社会分工的进一步精细化,计算机学科新知识、新领域层出不穷,多学科交叉与融合的计算机学科新形态正逐渐形成。2012年,国家教育部公布的《普通高等学校本科专业目录(2012年)》中将计算机类专业分为计算机科学与技术、软件工程、网络工程、物联网工程、信息安全、数字媒体技术等专业。

随着国家信息化步伐的加快和我国高等教育逐步走向大众化,计算机类专业人才培养不仅在数量的增加上也在质量的提高上对目前的计算机类专业教育提出更为迫切的要求。社会需要计算机类专业的教学内容的更新周期越来越短。相应的,我国计算机类专业教育也将改革的目标与重点聚焦于如何培养合能够适应社会经济发展需要的高素质工程应用型人才。

作为应用型地方本科院校,南京晓庄学院计算机类专业在多年实践中,逐步形成了行知“教学做合一”思想与国际工程教育理念相融合的独具晓庄特色的工程教育新理念。学生在社会生产实践的“做”中产生专业学习需求和形成专业认同,在“做”中增强实践能力和创新能力,在“做”中生成和创造新知识,在“做”中涵养基本人格和公民意识;同时学生应遵循工程教育理念,标准地“做”,系统地“做”,科学地“做”,创造地“做”。

实践环节是应用型本科院校人才培养的重要手段之一,是应用型人才培养目标得以实现的重要保证。当前市场上一些实践教材导向性不明显,可操作性不强,系统性不够,与社会生产实际联系不紧密。总体上来说没有形成系列,同一专业的不同实践教材重复较多,且教材之间的衔接不够。

《教育部关于“十三五”普通高等教育本科教材建设的若干意见(教高[2011]05号)》要求重视和发挥行业协会和知名企业在教材建设中的作用,鼓励行业协会和企业利用其具有的行业资源和人才优势,开发贴近经济社会实际的教材和高质量的实践教材。南京晓庄学院计算机类专业积极开展校企联合实践教材建设工作,与国内多家知名企业共同规划建设“应用型本科院校计算机类专业校企合作系列教材”。本系列教材是在计算机学科和计算机类专业课程体系建设基本成熟的基础上,按照国际工程教育认证的相关要求由我校优秀教学科研骨干和行业企业专家通力合作而成的,力求充分体现

科学性、先进性、工程性。本系列教材在规划编写过程中体现了如下一些基本组织原则和特点。

1. 贯彻了国际工程教育新理念。教材内容的组织和案例的甄选充分考虑了复杂工程背景和宏大工程视野下的工程项目组织、实施和管理,注重强化具有团队协作意识、创新精神等优秀人格素养的卓越工程师的培养。

2. 体现了计算机学科发展趋势和技术进步。教材内容适应社会对现代计算机工程人才培养的需求,反映了基本理论和原理的综合应用,反映了教学体系的调整和教学内容的及时更新,注重将数据处理等相关前沿技术的新成果、新应用纳入教材内容。

3. 反映了计算机类专业改革和人才培养需要。教材规划以 2012 年教育部公布的新专业目录为依据,正确把握了计算机类专业教学内容和课程体系的改革方向。在教材内容和编写体系方面注重了学思结合、知行合一和因材施教,强化了以适应社会需要为目标的教学内容改革,由知识本位转向能力本位,体现了知识、能力、素质协调发展的要求。

4. 结合了行业企业的技术优势和项目资源。教材采用校企联合开发和建设的模式,充分利用行业专家、企业工程师和项目经理的项目组织、管理、实施经验的优势,将企业实际实施的工程项目分解为若干可独立执行的案例,注重了问题探究、案例讨论、项目参与式教育教学方式方法的运用。

5. 突出了应用型本科院校基本特点。教材内容以适应社会需要为目标,突出“应用型”的基本特色,围绕培养目标,以工程应用为背景,通过理论与实践相结合,重视学生的工程应用能力的培养,增强学生对技能的应用。

相信通过这套“应用型本科院校计算机类专业校企合作实训教材系列”的规划出版,能够在形式上和内容上显著提高我国应用型本科院校计算机类专业实践教材的整体水平,继而提高计算机类专业人才的培养质量,培养出符合经济社会发展需要和产业需求的高素质工程应用型人才。

系列教材编委会

Java EE 技术经过多年的发展已日趋成熟,使用 Java EE 技术进行项目开发的企业和工程师越来越多。因此,计算机类专业的学生可以通过掌握 Java EE 技术进一步提升就业竞争力。

Java EE 技术建立在开源软件基础之上,它是许多软件开发工程师在开发实践中不断摸索提炼出的技术。Java EE 技术所包含的知识点非常庞杂,如何指导缺乏软件项目开发经验的学生或初学者较快地理解并掌握 Java EE 不是一件容易的事。当前,尽管很多高校开设了 Java EE 课程,但教学效果很不理想,合适教材的缺乏就是原因之一。问题主要体现在:第一,目前介绍 Java EE 技术的教材更注重知识的传授,在能力培养方面有所欠缺;第二,教材所举案例比较零散,缺少来源于企业的真实案例,不利于工程实践知识的传授和能力的培养;第三,大多数已有教材包含的知识点对于没有任何开发经验的大学生以及初学者而言难度偏大。因此,编写出符合应用型本科高校计算机技术和软件工程课程教学特点和需求的教材已刻不容缓。

本书的特点通俗易懂、实用性强以及综合案例来源于企业真实开发项目。编写人员由国内知名企业资深项目架构师和有丰富开发经验的高校教师组成。本书在内容的选取及章节的安排上相对于目前已有的教材做了一定的调整,删减了一些对于初学者而言难以理解的内容。本书前 6 章内容主要包括:Java EE 简介、Web 编程基础、Java Web 编程基础、Struts 2 应用、Hibernate 和 MyBatis 应用以及 Spring 应用。每个知识点都从最简单的例子着手,一步一步引导读者学习和实现这些案例。读者在案例实现过程中对涉及的知识点有了初步认识,即先达到“知其然”,教材最后通过“教学管理系统”、“教育资源网络平台”和“高校工程教育认证体系”这三个来源于企业的真实案例使读者进一步掌握和运用所学知识,最后达到“知其所以然”。经过整个课程的系统学习,读者不仅掌握了相关专业知识,并且对企业项目实际的开发流程有了一定的了解。

本书的完成得益于许多老师、学生以及合作企业的积极参与。其中,富士康南京软件有限公司的孙杰等工程师提供了部分素材资料,南京晓庄学院信息工程学院的徐家喜、候青、朱杰、李青等各位老师以及魏笑笑、杜瑞和周楠等学生参与了部分章节的案例代码调试和文字内容的编写工作,李青和包依勤老师参与了教材的审阅工作,在此一并向他们表示感谢。

本书第 1~6 章部分内容是对互联网资料进行收集、整理和改编的结果,网站来源

包括软件官方网站或软件教程网等。由于本书的大多数知识点和实验内容是建立在开源软件的基础上,而开源软件最大的特点就是广大开源软件爱好者的无私奉献。因此,本人在这里表示对他们的敬意和感谢。

由于作者水平有限,疏漏和错误在所难免,敬请广大师生、读者批评指正。意见和建议可反馈至邮箱: xz_wang@163.com。

本教材提供用于教师教学的配套光盘,光盘包含了教材中所有案例的源代码和开发工具。

编 者

2017年12月

第 1 章 Java EE 简介	1
1.1 Java EE 应用概述	1
1.1.1 Java EE 应用的四层结构	2
1.1.2 Java EE 应用的体系结构优点	2
1.2 Java EE 的轻型框架简介	3
1.2.1 Struts 框架	3
1.2.2 ORM 框架	4
1.2.3 Spring 框架	4
1.2.4 Spring MVC 框架	4
1.3 JSP 的开发环境搭建	5
1.3.1 JDK 的安装设置	5
1.3.2 Tomcat 安装设置	6
1.3.3 Eclipse 的安装与设置	8
1.4 应用实例	10
第 2 章 Web 编程基础	12
2.1 HTML 语言	12
2.1.1 HTML 基本概念	12
2.1.2 HTML 的标签	13
2.1.3 表格与框架	13
2.2 JavaScript 语言	16
2.2.1 JavaScript 简介	16
2.2.2 JavaScript 和 Java 的区别	17
2.2.3 JavaScript 基本数据结构	18
2.2.4 表达式和运算符	20
2.2.5 事件驱动及事件处理	22
2.2.6 基于对象的 JavaScript 语言	26
2.3 jQuery 基础	35

2.4	HTML5 语言	44
2.4.1	HTML5 简介	44
2.4.2	HTML5 视频	44
2.4.3	HTML5 音频	45
2.4.4	HTML5 Canvas	46
2.4.5	HTML5 Web 存储	47
2.4.6	HTML5 新的 Input 类型	49
2.4.7	HTML5 表单元素	50
第 3 章	Java Web 编程基础	52
3.1	JSP 简介	52
3.1.1	JSP 特点	52
3.1.2	JSP 基本语法	52
3.2	JSP+JavaBean 应用实例	69
3.2.1	应用举例 1	69
3.2.2	应用举例 2	71
3.3	Servlet	77
3.3.1	Servlet 简单实例	78
3.3.2	Servlet 的生命周期	81
3.3.3	JSP 与 Servlet 的关系	83
3.4	JSP+Servlet+JavaBean 应用	84
3.4.1	应用举例 1	84
3.4.2	应用举例 2	86
3.5	JDBC 基本概念	93
第 4 章	Struts 2 概述及基本应用	97
4.1	Struts 2 概述	97
4.1.1	MVC 思想概述	97
4.1.2	Struts 2 的基本流程	99
4.2	Struts 2 简单实例开发及工作流程	100
4.2.1	Struts 2 简单实例开发	100
4.3	Struts 2 的工作流程及文件详解	106
4.3.1	Struts 2 的工作流程	106
4.3.2	Struts 2 中主要文件详解	107
4.4	Struts 2 标签库简介	112
4.4.1	Struts 2 标签库概述	113
4.4.2	Struts 2 标签语法	113
4.4.3	表单标签	114
4.4.4	非表单标签	117

4.4.5	数据标签	119
4.4.6	控制标签	122
4.5	Struts 2 数据验证	126
4.5.1	Validate 方法	126
4.5.2	Validation 框架	131
4.6	拦截器	134
4.6.1	拦截器概述	134
4.6.2	拦截器的使用方法	135
4.6.3	实现自定义拦截器	138
4.7	文件上传	144
4.7.1	上传单个文件	144
4.7.2	上传多个文件	147
4.8	Struts 2 应用实例	150
第 5 章	Hibernate 和 MyBatis	157
5.1	ORM 简介	157
5.2	Hibernate 体系结构	158
5.3	Hibernate 应用实例	159
5.4	文件作用详解	162
5.5	Hibernate 核心接口	166
5.6	HQL	169
5.7	Hibernate 关系映射	170
5.8	Hibernate、Struts 2 应用案例	175
5.9	Mybatis 简介及应用	180
5.9.1	MyBatis 简介	180
5.9.2	MyBatis 工作流程	182
5.9.3	MyBatis 基本要素	182
5.9.4	应用示例	185
第 6 章	Spring 应用	189
6.1	Spring 概述	189
6.2	简单工厂模式	190
6.3	依赖注入	193
6.4	Spring 注入方式	196
6.4.1	设值注入	196
6.4.2	构造注入	198
6.5	Spring 核心接口及基本配置	199
6.5.1	Spring 核心接口	199
6.5.2	Spring 基本配置	200

6.5.3 Spring 容器中的 Bean	201
6.6 Spring 的 AOP	202
6.6.1 AOP 简介	202
6.6.2 代理机制	202
6.7 Struts 2、Spring 和 Hibernate 应用案例	208
第 7 章 Struts 2、Mybatis 和 Spring 整合应用——教务管理系统	213
7.1 项目简介	213
7.2 技术架构	215
7.3 项目创建流程(以创建培养方案为例)	217
第 8 章 Spring3 MVC 和 Hibernate 整合应用——教育资源平台开发实例	238
8.1 项目简介	238
8.2 两个项目实现技术比较	244
8.3 项目创建流程(以用户注册为例)	245
附录 A	260
A.1 测试用例说明文档	260
A.2 [STORY. eTraining. 001. 01]—[需求库管理]	262
附录 B	265
B.1 教育资源平台数据字典	265
参考文献	274



【微信扫码】
拓展阅读

第1章 Java EE 简介

学习目标

1. 了解 Java EE 体系结构
2. 了解 Java EE 体系结构的优点
3. 了解 Java EE 主流的轻型框架
4. JDK、Tomcat 和 Eclipse 安装配置

Java 是一种通用、并行、基于类的且面向对象的程序设计语言,是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台(即 Java SE, Java EE, Java ME)的总称。Java 平台由 Java 虚拟机(JVM)和 Java 应用程序接口(API)构成。Java 语言与 C、C++ 语言有相似之处,也有很大差别。Java 语言略去了 C、C++ 的一些特性而引入了其他语言的一些思想,Java 语言风格较为接近 C#。

Java 语言是强类型定义语言,这有助于编程人员很快发现问题,因为程序编译时就可以检测出类型错误。Java 语言也是静态语言,通过编译把程序源代码按 JVM 定义规范转换成独立于机器的字节代码,所以 Java 程序无需重新编译便可在不同类型的计算机上执行,即“编写一次,到处运行”。

Java 语言在内存管理、多线程等方面提供了相对简单的管理模式,使得程序员更容易学习和掌握。

综上所述,使用 Java 语言开发更快捷、方便,开发出的软件易于维护与扩展。因此,Java 语言在计算机的各种平台、操作系统,以及手机、移动设备等方面均得到广泛的应用。

1.1 Java EE 应用概述

目前,Java 平台有 3 个版本,它们分别是适用于小型设备和智能卡的 Java Micro 版(Java Micro Edition, Java ME)、适用于桌面系统的 Java 标准版(Java Standard Edition, Java SE)、适用于创建服务器应用程序和服务的 Java 企业版(Java Enterprise Edition, Java EE)。Java EE 是一套开发、部署和管理相关的复杂问题的体系结构,使用 Java 为主要编程语言,减轻企业开发困难,提供一系列模式的解决方案。Java EE 规范建立在 Java 编程语言基础之上,更适应企业的网络需求,保留了标准版的很多优点。

在新的技术需求下,Java EE 作为不同于以往的技术架构,规范了整个系统的开发流程,为开发者们提供了一套全新的技术框架。Java EE 的全称叫作 Java 企业版(Java Enterprise Edition)技术规范与指南,里面包含了大量的具体服务架构、组件及技术,有着共同的

标准及规格。Java EE 通过减少 XML 配置和简化 JAR 包以及使用更多的 POJO 和注解的设计方式来提高项目的开发效率。Java EE 使得依赖于不同商业平台的系统有了较强的兼容性,为企业级的系统开发和运行提供了强有力的保证。

1.1.1 Java EE 应用的四层结构

(1) 运行在客户端机器上的客户层:负责与用户直接交互。Java EE 支持多种客户端,可以是 Web 浏览器,也可以是专用的 Java 客户端程序。

(2) 运行在 Java EE 服务器上的表示层:该层利用 Java EE 中的 JSP 与 Servlet 技术,响应客户端的请求,并可向后访问业务逻辑组件。

(3) 运行在 Java EE 服务器上的业务逻辑层:主要封装了业务逻辑,完成复杂计算,提供事务处理、负载均衡、安全、资源连接等基本服务。

(4) 运行在 EIT(Enterprise Information Tier)层服务器上的企业信息层:该层包括了数据库系统、文件系统等。

Java EE 应用的四层结构如图 1-1 所示。

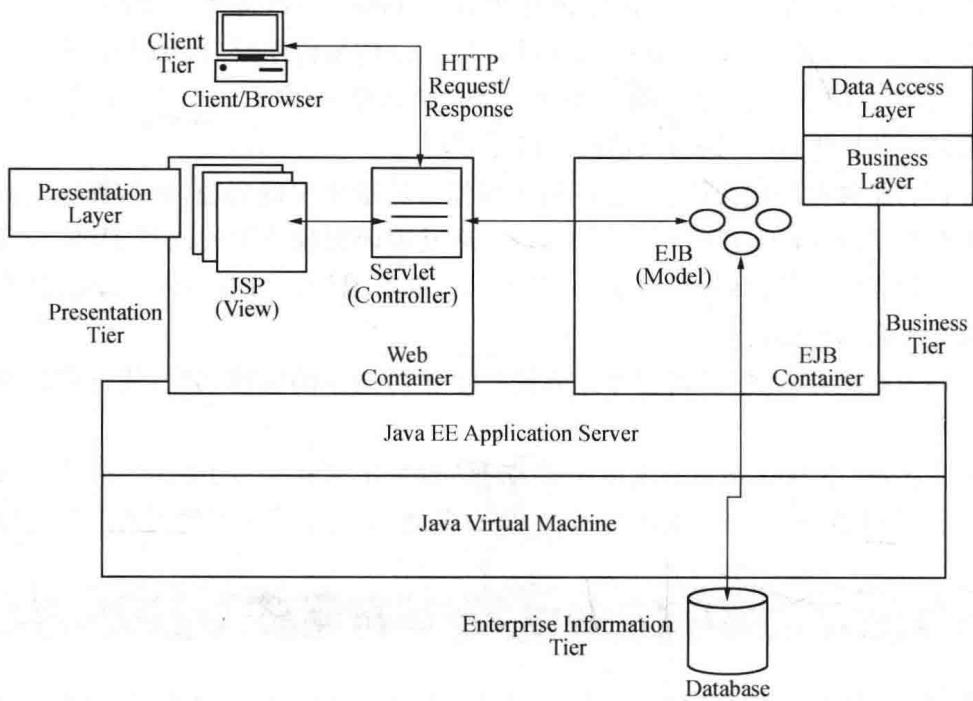


图 1-1 Java EE 四层结构

1.1.2 Java EE 应用的体系结构优点

1. 部署代价廉价

Java EE 体系结构提供了中间层集成框架以满足无需太多费用而又需要高可用性、高可靠性和可扩展性的应用需求。降低了开发多层应用的费用和复杂性,同时提供对现有应用程序集成强有力支持。

2. 开发高效

允许公司把一些通用的、很繁琐的服务端任务交给中间件供应商去完成。这样开发人员可以集中精力在如何创建商业逻辑上,从而可大大缩短开发时间。中间件供应商一般提供以下中间件服务:

- (1) 状态管理服务
- (2) 持续性服务
- (3) 分布式共享数据对象 Cache 服务

3. 支持异构环境

基于 Java EE 的应用程序不依赖任何特定操作系统、中间件、硬件,只需开发一次就可部署到各种平台。Java EE 标准允许客户下载与 Java EE 兼容的第三方组件,把它们部署到异构环境中。

4. 可伸缩

Java EE 平台提供了广泛的负载均衡策略,它能消除系统中的瓶颈,允许多台服务器集成部署,从而实现高度可伸缩。

1.2 Java EE 的轻型框架简介

软件开发框架将软件应用中的共性功能抽象出来,预先形成封装好的、与底层无关的、简单易用的接口。框架中通常还集成了很多类库,软件开发人员可以根据需要有选择性地调用或重写,从而完成对数据源、网络、系统等底层框架的访问。同时,软件开发框架并不完全等同于类库。框架除了提供类库外,还提供了“控制反转”的功能。在使用框架开发软件的过程中,对象实例化及方法调用是由框架实现的。其根本目的还是缩短开发周期,提高开发效率,提高软件的健壮性和可重用性。

Struts、Spring、Hibernate/Mybatis 都是当前 J2EE 开发 Web 应用的主流框架,支持者众多,下面先来简要介绍这三个框架,让大家有一个初步印象。

1.2.1 Struts 框架

Struts 是一种基于 MVC 经典设计模式的开放源代码的应用框架,也是目前 Web 开发中比较成熟的一种框架。“Struts”的含义即为专业应用开发提供一种“无形的支撑”,它通过把 Servlet、JSP、JavaBean、自定义标签和信息资源等 Java 平台的各种元素整合到一个统一的框架中,为 Web 开发提供具有高可配置性的 MVC 开发模式。

Struts 体系结构实现了 MVC 设计模式的概念,它将 Model、View 和 Controller 分别映射到 Web 应用中的组件。Mode 提供了应用程序的核心功能,它包含应用程序的数据和商业逻辑,并且封装了应用程序的状态。View 是由 JSP 和 Struts 提供的自定义标签(JSTL、JSF)来实现。Controller 负责流程控制,充当 Model 和 View 之间的桥梁。由 ActionServlet 负责读取 struts-config.xml,并使用 ActionMapping 来查找对应的 Action。Struts 的体系结构与工作原理如图 1-2 所示。

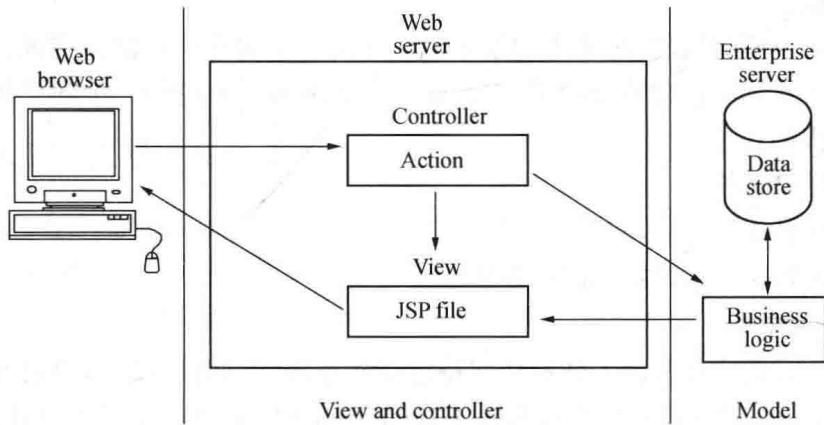


图 1-2 Struts 的体系结构与工作原理

1.2.2 ORM 框架

对目前的 Java EE 信息化系统而言,通常采用面向对象分析和面向对象设计的过程。系统从需求分析到系统设计都是按面向对象方式进行。通常,在一个程序应用中,都需要传递并持久化对象。传统方法是打开 JDBC 连接,接着创建 SQL 语句并把所需的参数值传递给它。若对象的参数较少时,这样做还比较容易,但当对象的参数很多时,实现和维护就很麻烦了。同时,这种实现方法也不符合面向对象的思想。因此,对象-关系映射(Object-relational mapping)应运而生。

ORM(Object-relational mapping)是一种基于 SQL 模式把对象模型映射到关系型数据模型的数据映射技术。Hibernate 和 Mybatis 都是面向 Java 环境的对象/关系映射工具,它可将对象模型表示的对象映射到基于 SQL 的关系数据模型中。ORM 把相应 Java 对象操作自动转换成 SQL 操作,程序开发者可以很容易地持久化 Java 对象。

1.2.3 Spring 框架

Spring 框架是 Rod Johnson 开发的,2003 年发布了 Spring 框架的第一个版本。Spring 是一个从实际开发中抽取出来的框架,因此它完成了大量开发中的通用步骤,从而大大提高了企业应用的开发效率。

Spring 为企业应用的开发提供了一个轻量级的解决方案。其中依赖注入、基于 AOP 的声明式事务管理、多种持久层的整合等最为人们关注。Spring 可以贯穿程序的各层之间,能够高效地组织应用程序中的各种中间层组件,但它并不是要取代那些已有的框架(如 Struts、Hibernate 等),而是以高度的开发性与它们紧密地整合,这也是 Spring 被广泛应用的原因之一。

1.2.4 Spring MVC 框架

Spring MVC 是 Spring Framework 的一个 Web 组件,已经融合在 Spring Web Flow 里面。Spring 框架提供了构建 Web 应用程序的全功能 MVC 模块。使用 Spring 可整合的 MVC 架构,可以选择是使用内置的 Spring Web 框架或是 Struts、JSF 等 Web 框架。通过

策略接口, Spring MVC 框架是高度可配置的, 而且包含多种视图技术, 例如 Java Server Pages(JSP)技术、Servlet 和 Tiles 等。Spring MVC 框架并不需要知道使用的视图, 所以不会强迫您只使用 JSP 技术。

Spring MVC 分离了控制器、模型对象、分派器以及处理程序对象的角色, 这种分离让它们更容易进行定制。易于同其他 View 框架(Tiles 等)无缝集成, 采用 IOC 便于测试。它是一个典型的教科书式的 MVC 构架, 而不像 Struts 等都是变种或者不是完全基于 MVC 系统的框架。它和 Tapestry 一样是一个纯正的 Servlet 系统, 这也是 Struts 所没有的优势。而且框架本身有代码, 看起来也不费劲, 比较简单更易理解。

1.3 JSP 的开发环境搭建

1.3.1 JDK 的安装设置

1. JDK 的下载和安装

JDK(Java Development Kit)即 Java 开发工具包, 进行 JSP 开发工作必须要有 JDK。本书使用的 jdk1.7.0_06, 可以在 <http://java.sun.com> 下载到文件为 jdk-jdk1.7.0_06-windows-i586-p.exe 的 Java 工具包。

2. JDK 的环境变量配置

在“我的电脑”上单击鼠标右键, 在弹出的快捷菜单中执行【属性】命令, 在弹出的对话框中选择【高级】选项卡, 单击【环境变量】按钮, 在打开对话框中添加如下的环境变量。

(1) 设置 JAVA_HOME 变量为 Java 的主目录 C:\Program Files\Java\jdk1.7.0_06

```
$ JAVA_HOME = C:\Program Files\Java\jdk1.7.0_06
```

(2) 把 Java 的 bin 目录路径 D:\jdk1.7.0_06\bin 添加到 PATH 环境变量中:

```
$ PATH = %JAVA_HOME%\bin;
```

(3) 设置 CLASSPATH 变量:

```
$ CLASSPATH = .; %JAVA_HOME%\lib\tools.jar; %JAVA_HOME%\lib
```

注意: 在两个环境变量的设置中, 都包含了一个 "." 的路径, 这个 "." 代表系统的当前路径, 如果没有增加该路径, 可能导致运行 Java 程序时, class 文件已在当前路径, 但在系统提供的文件中找不到该文件。

3. JDK 测试

(1) 用文本编辑器写一个简单的 Java 程序

```
public class HelloWorld {
    public static void main(String args[]) {
        System.out.println("Hello World!");
    }
}
```


这个例子就是著名的“Hello World”，它的功能就是显示“Hello World”。

注意：该文件名称必须为“HelloWorld.java”，大小写也区分。细心的朋友会注意到就是和 public class 后的名字一样的。

(2) 编译：在 DOS 命令提示符下执行（注意大小写）

```
javac HelloWorld.java
```

如果正常的话，将生成 HelloWorld.class 文件。

(3) 运行：在 DOS 命令提示符下执行（注意大小写）

```
java HelloWorld
```

1.3.2 Tomcat 安装设置

1. Tomcat 下载

Apache Jakarta 项目组开发的基于 GPL 自由软件协议的 JSP 引擎，配合 JDK 就可以搭建起一个最简单的 JSP 试验平台。在 <http://tomcat.apache.org/> 下载版本 7.0.19，下载的文件为 apache-tomcat-7.0.19-windows-x86.zip。

2. 安装 Tomcat

直接解压 apache-tomcat-7.0.19-windows-x86.zip 文件，双击 startup.bat 文件，启动 Tomcat。启动 Tomcat 之后，打开浏览器，在地址栏输入 <http://localhost:8080>，然后按回车键，浏览器出现如图 1-3 所示界面，即表示 Tomcat 安装成功。

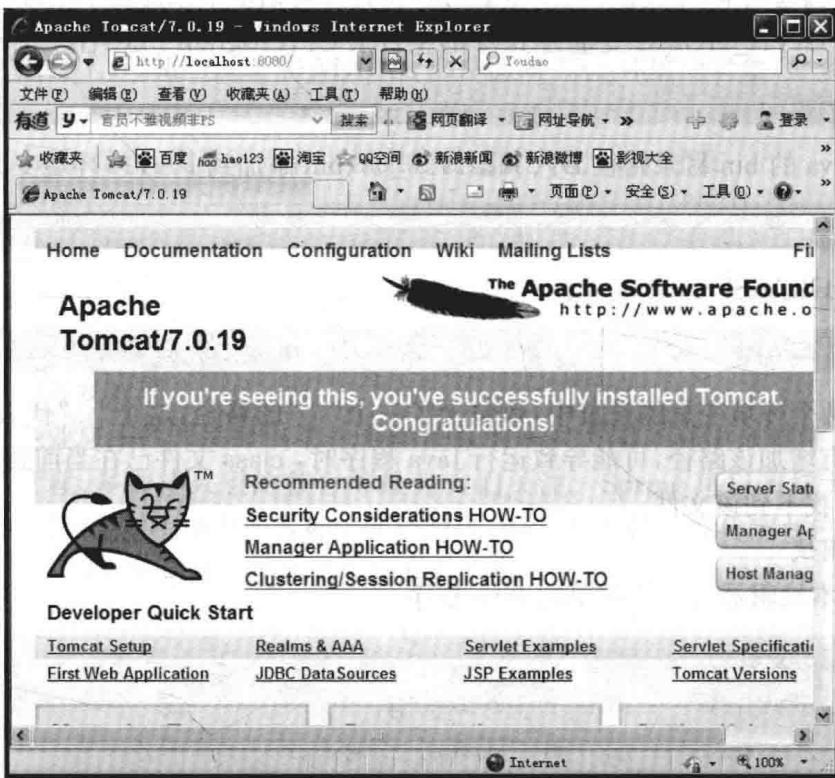


图 1-3 Tomcat 启动