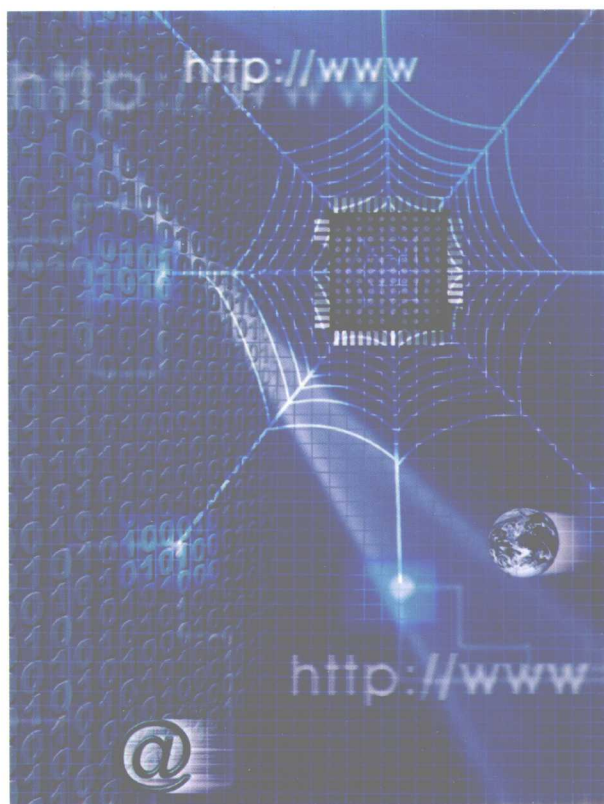


JSP实用简明教程

(第2版)

- ◆ JSP相关知识
- ◆ Java语言基础
- ◆ 应用开发体系与环境配置
- ◆ 动态网页设计基础
- ◆ JSP内置对象与文件操作
- ◆ JSP访问数据库
- ◆ Servlet与JavaBeans编程
- ◆ 在线书店实例



钟德源 林国 程志艳 编著



清华大学出版社

高等学校计算机应用规划教材

JSP 实用简明教程

(第 2 版)

钟德源 林国 程志艳 编著

清华大学出版社

北 京

内 容 简 介

本书主要讲述 JSP 动态网站开发的相关知识, 内容包括 JSP 相关技术概述、Java 语言基础、JSP 的内
置对象、JSP 开发环境的搭建、JSP 动态网页设计基础、JSP 的文件操作、JSP 访问数据库、JSP 与 Servlet
编程、JSP 与 JavaBeans 编程, 最后以一个典型案例详细讲解了 JSP 各种技术的综合应用。

本书采用基础知识结合典型案例的方法进行技术讲解, 并附赠习题, 以便读者更好地巩固所学的 JSP
技术知识。

本书可作为大学计算机及相关专业 JSP 课程的教材, 也适合自学者及网站开发人员参考使用。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

JSP 实用简明教程(第 2 版)/钟德源, 林国, 程志艳 编著. —北京: 清华大学出版社, 2009.5
(高等学校计算机应用规划教材)

ISBN 978-7-302-20085-7

I. J… II. ①钟…②林…③程… III. JAVA 语言—主页制作—程序设计—高等学校—教材
IV. TP393.092

中国版本图书馆 CIP 数据核字(2009)第 064417 号

责任编辑: 刘金喜 鲍 芳

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 李红英

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: 北京国马印刷厂

经 销: 全国新华书店

开 本: 185×260 印 张: 23.25 字 数: 580 千字

版 次: 2009 年 5 月第 2 版 印 次: 2009 年 5 月第 1 次印刷

印 数: 1~5000

定 价: 35.00 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: 029887-01

前 言

Java Server Pages (JSP)是基于 Java 的技术,用于创建可支持跨平台及跨 Web 服务器的动态网页。JSP 技术的设计目的是使得构造基于 Web 的应用程序更加容易和快捷,而这些应用程序能够与各种 Web 服务器、应用服务器、浏览器和开发工具共同工作。在 Sun 正式发布 JSP 之后,这种新的 Web 应用开发技术很快引起了人们的关注,目前已经成为开发动态网页的主流技术之一。

本书是一本关于 JSP 课程的大学教材,书中囊括了使用 JSP 进行程序开发的基础知识,同时在讲解中结合了大量实用的示例,并在每章后面提供了习题。

本书分为 10 章。书中各章节遵循由浅入深的方式进行讲解,使读者更快速地掌握 JSP 编程知识,掌握具体方法和使用技巧。

本书内容

第 1 章: JSP 概述。包括了常用的目前主流的 Java 技术、JSP 技术和目前使用 JSP 开发应用程序的两种体系结构。

第 2 章: Java 语言基础。介绍了 Java 语言的特点、数据类型、运算符、条件语句、循环语句、分支语句、强制类型的转换、Java 类技术基础、Java 类高级技术和多线程技术。

第 3 章: JSP 应用开发体系与环境配置,介绍了 Web 应用程序理论基础、JSP 实用原理和运行机制、JDK 的安装、Eclipse 和 MyEclipse 的安装与配置。

第 4 章: JSP 动态网页设计基础。介绍了 HTML 和 CSS 的语法、JSP 的基础知识,包括 JSP 页面结构、JSP 注释语句、JSP 变量和方法的声明、JSP 指令元素及动作元素。

第 5 章: JSP 主要内置对象。介绍了 JSP 的九大内置对象,包括 request 对象、response 对象、page 对象、session 对象、application 对象、out 对象、config 对象、exception 对象和 pageContext 对象。

第 6 章: JSP 中的文件操作。介绍了异常处理机制、JSP 的输入和输出流、File 类概述及如何读取、写入文件。

第 7 章: JSP 访问数据库。介绍了 JDBC 概念,然后介绍了使用 JDBC-ODBC 访问数据库、使用 JDBC 驱动程序访问数据库、对数据库的增删改查操作、对 JSP 页面进行分页、与 Excel 组件的结合等。

第 8 章: JSP 与 Servlet 编程。介绍了 Servlet 概念、Servlet 的常用方法、Servlet 的生命周期、Servlet 的常用类和接口。

第 9 章: JSP 与 JavaBeans 编程。介绍了 JavaBeans 的概念、属性和方法,如何在 JSP 中使用 JavaBeans、通过 JavaBeans 来操作数据库。

第 10 章: 在线书店。介绍了在线书店的需求分析、系统分析设计、界面设计和代码实现。

本书特点

- 由浅入深，循序渐进：本书按照由浅入深、由易到难的学习过程安排课程内容，概念讲解清晰，技术分析透彻，使读者逐步掌握 JSP 相关技术。
- 案例典型，操作性强：本书在介绍关键知识点时，都配有一个典型案例，具体说明知识点的使用方法，让读者亲自试一试，加深对知识点的理解和掌握。
- 注释详尽，可读性强：本书对实例代码统一编号，详细注解，增强了可读性，易学易用。
- 配套丰富，便于教学：本书提供案例源代码和课后习题，便于读者实践、练习。为便于教师教学，还配备了详尽的电子教案。这些资源可通过 <http://www.tupwk.com.cn/downpage> 下载。

学时安排

本课程总学时为 76 学时，各章学时分配见下表(供参考)：

课 程 内 容	学 时 数			
	合 计	讲 授	实 验	机 动
第 1 章 JSP 概述	2	2		
第 2 章 Java 语言基础	8	4	4	
第 3 章 JSP 应用开发体系与环境配置	6	4	2	
第 4 章 JSP 动态网页设计基础	10	6	4	
第 5 章 JSP 主要内置对象	8	4	4	
第 6 章 JSP 中的文件操作	8	4	4	
第 7 章 JSP 访问数据库	10	6	4	
第 8 章 JSP 与 Servlet 编程	6	4	2	
第 9 章 JSP 与 JavaBeans 编程	6	4	2	
第 10 章 在线书店	12	4	4	4
合计	76	42	30	4

本书由钟德源、林国、程志艳编著，此外，蓝荣香、王昊亮、喻波、马天一、魏勇、郝荣福、孙明、李大宇、武思宇、牟博超、李彬、付鹏程、高翔、朱丽云、张巧玲、李辉、李欣、柏宇和刘湘等同志在整理材料方面给予了编者很大的帮助，在此，编者对他们表示衷心的感谢。

在本书的编写过程中，借鉴了许多现行教材的宝贵经验，在此，谨向这些作者表示诚挚的感谢。由于时间仓促，加之编者水平有限，书中难免有谬误或不足之处。敬请广大读者批评指正。

编 者
2009 年 3 月

目 录

第 1 章 JSP 概述	1
1.1 Java 技术	1
1.1.1 Java Beans	1
1.1.2 JDBC	2
1.1.3 J2EE	3
1.1.4 Java Servlet	6
1.1.5 Struts	7
1.1.6 Spring	8
1.2 JSP 技术	9
1.2.1 JSP 技术概述	9
1.2.2 Web 应用开发的 JSP 技术	10
1.2.3 JSP 页面	11
1.2.4 JSP 页面的应用模型	13
1.3 用 JSP 开发 Web 站点的几种方式	14
1.4 理解 JSP 的两种体系结构	16
1.4.1 Model 1	16
1.4.2 Model 2	16
1.5 小结	17
1.6 习题	18
1.6.1 填空题	18
1.6.2 选择题	18
1.6.3 简答题	19
第 2 章 Java 语言基础	20
2.1 Java 语言的特点	20
2.2 Java 的数据类型	22
2.2.1 简单类型	22
2.2.2 数组	26
2.3 运算符	28
2.3.1 算术运算符	29

2.3.2 整数位运算符	30
2.3.3 关系运算符	32
2.3.4 布尔逻辑运算符	33
2.3.5 赋值运算符	34
2.3.6 条件运算符	35
2.3.7 运算符优先级	35
2.4 条件语句	36
2.5 循环语句	38
2.5.1 while 语句	38
2.5.2 do-while 语句	38
2.5.3 for 循环	39
2.5.4 循环退出语句	39
2.6 分支语句	41
2.7 Java 中的强制类型转换	42
2.7.1 简单数据类型的强制转换	42
2.7.2 简单数据类型和对象之间的强制转换	43
2.8 Java 类技术基础	43
2.8.1 类的声明和类体	43
2.8.2 成员的声明	44
2.8.3 方法调用	45
2.8.4 构造函数	46
2.8.5 重载	47
2.8.6 继承	48
2.9 Java 高级类技术	49
2.9.1 抽象类	50
2.9.2 匿名类	51
2.9.3 包和接口	52
2.10 多线程技术	56
2.10.1 多线程的定义	56
2.10.2 线程优先级	56

2.10.3 线程同步	57	4.1.7 表单	96
2.11 小结	57	4.2 CSS 语法	102
2.12 习题	58	4.2.1 CSS 简介	102
2.12.1 填空题	58	4.2.2 CSS 基本语法	104
2.12.2 选择题	58	4.2.3 CSS 基本属性	108
2.12.3 上机题	58	4.3 JSP 页面的基本结构	113
第 3 章 JSP 应用开发体系与		4.4 JSP 注释	114
环境配置	59	4.4.1 隐藏注释	114
3.1 Web 应用程序理论基础	59	4.4.2 HTML 注释	115
3.1.1 Internet 技术概述	59	4.4.3 Java 注释	116
3.1.2 HTTP 工作原理	62	4.5 变量、方法的声明	117
3.2 JSP 实现原理和运行机制	63	4.5.1 变量、方法的声明和表达式	117
3.2.1 JSP 实现原理	63	4.5.2 表达式	119
3.2.2 JSP 运行机制	65	4.6 JSP 指令	119
3.3 JSP 环境配置	68	4.6.1 include 指令	119
3.3.1 JDK 的安装与配置	69	4.6.2 Page 指令	121
3.3.2 Tomcat 的安装与配置	71	4.6.3 taglib 指令	123
3.3.3 Eclipse+MyEclipse 的配置	72	4.7 JSP 动作(Action)语法	124
3.3.4 使用 Eclipse+MyEclipse		4.7.1 <jsp:include>动作	124
开发 JSP 程序	73	4.7.2 <jsp:useBean>动作	125
3.4 小结	78	4.7.3 <jsp:setProperty>动作	126
3.5 习题	79	4.7.4 <jsp:getProperty>动作	128
3.5.1 填空题	79	4.7.5 <jsp:forward>动作	129
3.5.2 选择题	79	4.7.6 <jsp:plugin>	130
3.5.3 上机题	79	4.8 JSP 脚本	132
第 4 章 JSP 动态网页设计基础	80	4.9 JavaScript 脚本	132
4.1 HTML 语法	80	4.10 小结	134
4.1.1 什么是 HTML 语言	80	4.11 习题	134
4.1.2 HTML 的基本标签	81	4.11.1 填空题	134
4.1.3 创建列表	86	4.11.2 选择题	135
4.1.4 图像、多媒体和超链接	88	4.11.3 上机题	135
4.1.5 表格	90	第 5 章 JSP 主要内置对象	137
4.1.6 框架	93	5.1 Request 对象	137
		5.1.1 Request 对象的方法	137

5.1.2 Request 对象程序应用实例	139
5.2 Response 对象	146
5.2.1 Response 对象的方法	146
5.2.2 Response 对象程序应用实例	148
5.3 Session 对象	151
5.3.1 Session 对象的方法	151
5.3.2 Session 对象程序应用实例	152
5.4 Out 对象	154
5.4.1 Out 对象的方法	155
5.4.2 Out 对象程序应用实例	156
5.5 PageContext 对象	157
5.6 Application 对象	157
5.6.1 Application 对象的方法	158
5.6.2 Application 对象程序 应用实例	158
5.7 Config 对象	160
5.7.1 Config 对象的方法	160
5.7.2 Config 对象程序应用实例	161
5.8 Exception 对象	162
5.8.1 Exception 对象的方法	162
5.8.2 Exception 对象程序 应用实例	163
5.9 小结	164
5.10 习题	164
5.10.1 填空题	164
5.10.2 选择题	165
5.10.3 上机题	166
第 6 章 JSP 中的文件操作	168
6.1 文件操作相关异常处理	168
6.1.1 异常处理概述	168
6.1.2 为什么要使用异常处理	169
6.1.3 异常类的层次	169
6.1.4 异常的处理	169
6.1.5 自定义异常	175
6.2 JSP 的输入和输出流	176
6.2.1 字节流	177
6.2.2 字符流	185
6.3 File 类概述	189
6.3.1 目录	190
6.3.2 创建目录	192
6.3.3 获取信息	193
6.4 在页面中读写文件中的数据	194
6.4.1 读取操作	194
6.4.2 写入操作	197
6.4.3 追加操作	200
6.4.4 在任意位置读取、写入操作	203
6.5 文件操作实例	203
6.5.1 文件的操作实例界面	203
6.5.2 程序清单	207
6.6 小结	211
6.7 习题	212
6.7.1 填空题	212
6.7.2 选择题	212
6.7.3 上机题	213
第 7 章 JSP 中访问数据库	215
7.1 JDBC 概述	215
7.1.1 什么是 JDBC	215
7.1.2 JDBC 与 ODBC	216
7.1.3 JDBC 两层模型和三层模型	217
7.1.4 JDBC 接口	218
7.2 使用 JDBC—ODBC 桥 访问数据库	221
7.2.1 创建一个 Access 型数据库	221
7.2.2 创建 ODBC 数据源	222
7.2.3 编写 JSP 程序	223
7.2.4 JDBC-ODBC 桥访问 Access 数据库分析	225

7.3 使用 JDBC 驱动程序	
访问数据库.....	227
7.3.1 下载并安装 JDBC 驱动程序.....	228
7.3.2 创建 SQL Server 数据库.....	230
7.3.3 编写 JSP 程序.....	231
7.3.4 JDBC 直接访问 SQL Server 数据库分析.....	233
7.4 查询记录.....	235
7.4.1 SQL 的查询语句.....	235
7.4.2 JSP 发送执行查询语句.....	236
7.5 更新记录.....	239
7.5.1 SQL 的更新语句.....	239
7.5.2 JSP 发送执行更新语句.....	240
7.6 添加记录.....	240
7.6.1 SQL 的插入语句.....	240
7.6.2 JSP 发送执行插入语句.....	241
7.7 删除记录.....	241
7.7.1 SQL 的删除语句.....	241
7.7.2 JSP 发送执行删除语句.....	242
7.8 分页显示记录.....	242
7.8.1 分页显示实例.....	242
7.8.2 代码实现分析.....	245
7.9 查询 Excel 电子表格.....	247
7.9.1 创建 Excel 文件.....	247
7.9.2 创建 Excel 数据源.....	247
7.9.3 编写 JSP 文件.....	248
7.9.4 访问 Excel 表格数据 实现分析.....	249
7.10 JSP 访问 MySQL 数据库.....	250
7.11 JSP 访问 SQL Server 2005 数据库.....	250
7.12 JSP 数据库应用实例.....	251
7.12.1 程序效果图.....	251
7.12.2 程序清单.....	253
7.13 小结.....	261
7.14 习题.....	262
7.14.1 填空题.....	262
7.14.2 选择题.....	262
7.14.3 上机题.....	262
第 8 章 JSP 与 Servlet 编程.....	263
8.1 Servlet 概述.....	263
8.1.1 什么是 Servlet.....	263
8.1.2 Servlet 的优点.....	264
8.1.3 Servlet 的用途.....	264
8.1.4 Servlet 与 JSP 的区别.....	265
8.1.5 Servlet 的方法.....	265
8.1.6 Servlet 的生命周期.....	269
8.1.7 Servlet 的类和接口.....	273
8.1.8 状态代码.....	277
8.2 编写一个简单的 Servlet.....	280
8.3 处理客户端提交的数据.....	281
8.4 返回数据到客户端.....	283
8.5 JSP+Servlet 编程实例.....	285
8.6 小结.....	292
8.7 习题.....	292
8.7.1 填空题.....	292
8.7.2 选择题.....	293
8.7.3 上机题.....	293
第 9 章 JSP 与 JavaBeans 编程.....	294
9.1 为什么要使用 JavaBeans.....	294
9.1.1 什么是 JavaBeans.....	294
9.1.2 JavaBeans 的种类.....	295
9.2 实现简单的 JavaBeans.....	295
9.3 在页面中使用 JavaBeans.....	299
9.3.1 JavaBeans 的属性.....	299
9.3.2 在 JSP 中使用 JavaBeans.....	303
9.4 通过 JavaBeans 实现数据库 的访问.....	312

9.4.1 连接数据库.....	313	10.3 代码实现.....	334
9.4.2 数据查询.....	314	10.3.1 通用模块.....	334
9.5 JSP 与 JavaBeans 编程实例.....	316	10.3.2 用户在线购书.....	337
9.6 小结.....	321	10.3.3 用户查看订单.....	344
9.7 习题.....	321	10.3.4 管理员登录.....	346
9.7.1 填空题.....	321	10.3.5 图书管理.....	346
9.7.2 选择题.....	321	10.3.6 订单管理.....	350
9.7.3 上机题.....	322	10.3.7 管理用户.....	354
第 10 章 在线书店.....	324	10.4 程序的运行.....	358
10.1 需求分析.....	324	10.5 小结.....	359
10.2 系统分析设计.....	325	10.6 习题.....	360
10.2.1 功能模块分析.....	325	10.6.1 填空题.....	360
10.2.2 数据库结构设计.....	326	10.6.2 上机题.....	360
10.2.3 界面设计.....	331		

第1章 JSP 概述

JSP(Java Server Pages)是由 Sun Microsystems 公司倡导的、许多公司参与一起建立的一种动态网页技术标准。JSP 技术的设计目的是使得构造基于 Web 的应用程序更加容易和快捷,而这些应用程序能够与各种 Web 服务器、应用服务器、浏览器和开发工具共同工作。在 Sun 正式发布 JSP 之后,这种新的 Web 应用开发技术很快引起了人们的关注,目前已经成为开发动态网页的主流技术之一。

本章学习目标:

- 了解常用的 Java 技术
- 理解 JSP 技术
- 了解 JSP 开发 Web 站点的几种方式
- 了解 JSP 的两种体系结构

1.1 Java 技术

1995 年美国 Sun 公司倡导和推出 Java 技术。Java 是一种简单的、完全面向对象的、跨平台且安全可靠的、主要面向 Internet 的开发工具。Java 技术包括 Java 语言和 Java Media APIs、Security APIs、Management APIs、Java Applet、Java RMI、JavaBeans、Java OS、Java Servlet、JDBC、JNDI 等,自从 1995 年正式问世以来,Java 的快速发展已经极大地影响了整个 Web 世界的发展。随着 Java Servlet 的推出,Java 在电子商务方面开始崭露头角,Java Server Page 技术的推出,更是让 Java 成为基于 Web 的应用程序的首选开发工具。

1.1.1 JavaBeans

JavaBeans 是 Java 的可重用组件。众所周知,ASP 通过 COM 来扩充复杂的功能,如文件上载、发送 E-mail 以及将业务处理或复杂计算分离出来成为独立可重复利用的模块,而 JSP 通过 JavaBeans 实现了类似的功能扩充。JSP 对于在 Web 应用中集成 JavaBeans 组件提供了很好的支持,例如程序员可以直接利用经测试和可信任的已有组件,避免了重复开发,这样既节省了开发时间,也为 JSP 应用带来了更多的可伸缩性。JavaBeans 组件还可以用来执行复杂的计算任务,或负责与数据库的交互以及数据提取等。

JavaBeans 是一个十分完整的组件模型,它能够支持标准组件技术中的各种特性,如属性(properties)、事件(events)、方法(method)和持续性(persistence),这和“对象”是十分相似的。设计一个 JavaBeans 分为以下几个步骤:

- 指定 Beans 的属性。
- 指定 Beans 所产生或响应的事件。
- 定义 Beans 应公开给其他 Bean 的属性、方法和事件。

JavaBeans 开发起来简单,又可以利用 Java 语言的强大功能,许多动态页面处理过程实际上被封装到了 JavaBeans 当中,所以 JSP 页面比传统的 ASP 或 PHP 页面更加简洁。

1.1.2 JDBC

JDBC 是用于执行 SQL 语句的 Java 应用程序接口,由一组用 Java 语言编写的类和接口组成。在 JSP 中将使用 JDBC 来访问数据库。JDBC 是一种规范,它让各数据库厂商为 Java 程序员提供标准的数据库访问类和接口,这样就使得独立于 DBMS 的 Java 应用程序的开发工具和产品成为可能。一般来说, JDBC 类文件和 Java 小应用程序存放在客户端,用来访问位于远程服务器上的数据库。

JDBC 通过数据库的 JDBC 驱动程序来访问数据库,当然需要数据库的支持,例如 Oracle 就有 JDBC 驱动程序。如图 1-1 所示为 JDBC 通过数据库的 JDBC 驱动程序来访问数据库的过程。

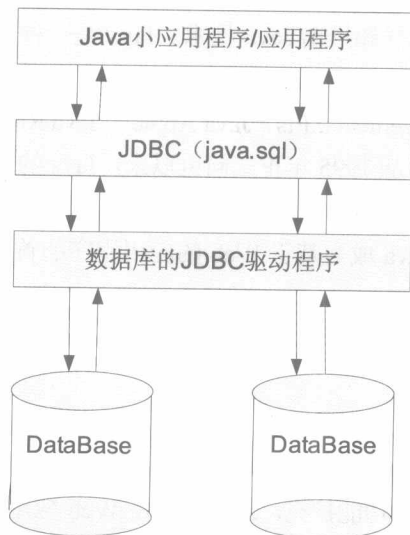


图 1-1 通过数据库的 JDBC 驱动程序来访问数据库

一般的 Java 开发工具都带有 JDBC-ODBC 桥驱动程序,这样只要是能够使用 ODBC 访问的数据库系统,也就能够使用 JDBC 访问,如图 1-2 所示。

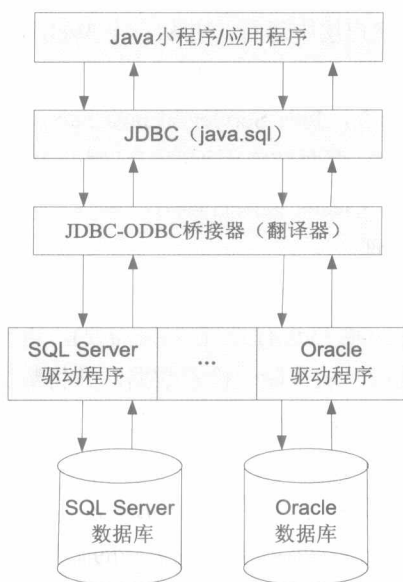


图 1-2 通过 JDBC-ODBC 桥驱动程序访问数据库

1.1.3 J2EE

随着电子商务和信息技术的快速发展，应用程序开发人员面临着许多新的压力：必须比以前更少的财力、更少的资源来更快地设计、建立和生产企业应用程序。

J2EE(Java 2 Platform Enterprise Edition)平台规范是一个由 SUN 公司定义的用于简化分布式企业级应用开发和部署的基于组件的模式。为了降低成本，加快企业应用程序的设计和开发，J2EE 平台提供了一个基于组件的方法来设计、开发、装配及部署企业应用程序，并提供了多层的分布式应用模型、组件再用、一致化的安全模型以及灵活的事务控制。因此程序员可以用更快的速度向市场推出创造性的客户解决方案，并且独立的、基于组件的 J2EE 解决方案不会被任何一个厂商的产品和 API 所束缚。

J2EE 平台由一整套服务(Services)、应用程序接口(APIs)和协议构成，它对开发基于 Web 的多层应用提供了功能支持。

1. J2EE 的层和组件

J2EE 使用多层的分布式应用模型，其应用逻辑按功能划分为组件，各个应用组件根据它们所在的层分布在不同的机器上。J2EE 分为 4 个层：客户层(Client Tier)、Web 层(Web Tier)、业务层(Business Tier)和企业信息系统层(Enterprise Information System Tier)。

客户层(Client Tier)运行在客户端机器上。应用于客户层的组件是“应用的客户组件(Application client components)”，它可以是基于 Web 方式的，即作为 Web 服务器的浏览器，也可以是基于传统方式的(非基于 Web 方式)，即独立的应用程序，可以完成瘦客户机无法完成的任务。

J2EE 应用程序可以是基于 Web 的，也可以是不基于 Web 的。在一个基于 Web 的 J2EE

应用程序中, 用户的浏览器在客户层中运行, 并从一个 Web 服务器上下载 Web 层中的静态 HTML 页面或者由 JSP 或 Servlets 生成的动态 HTML 页面。

Web 层运行在 Web 服务器上, Java Servlet 和 Java Server Pages(JSP)是 Web 层组件。调用 Servlet 或者 JSP 页面的 HTML 页面在应用程序组装时与 Web 组件打包在一起。与客户层相似, Web 层可能包括一个 JavaBeans 类来管理用户输入, 并将输入发送到在业务层中运行的 Enterprise Beans 类来进行处理。

业务层运行在 EJB 服务器上, Enterprise JavaBeans(EJB)是业务层组件。一个 Enterprise Bean 从客户程序处接收数据, 对数据进行处理(如果需要), 再将数据发送到企业信息系统层存储。一个 Enterprise Bean 还可以从存储中检索数据, 并将数据送回客户程序。运行在业务层的 Enterprise Beans 依赖于容器来为诸如事务、生命期、状态管理、多线程及资源存储池提供通常非常复杂的系统级代码。

Enterprise Bean 分为 3 类: 会话(Session)Beans、实体(Entity)Beans 和消息驱动(Message-driven)Beans。会话 Bean 表示与客户端程序的临时交互; 实体 Bean 表示数据库的表中一行永久的记录; 消息驱动 Bean 结合了会话 Bean 和 JMS 的消息监听器的特性, 允许一个业务层组件异步接收 JMS 消息。

业务层经常被称作 Enterprise JavaBeans(EJB)层。业务层和 Web 层一起构成了 3 层 J2EE 应用的中间层, 而其他两层是客户层和企业信息系统层。

企业信息系统层运行在 EIS 服务器上, 这一层上运行企业信息系统软件。

2. 设计用户界面和引擎(Designing the User Interface and Engine)

在为 J2EE 应用程序设计用户界面和后端引擎时, 需要考虑该程序是否要基于 Web, 还要考虑平台配置、下载速度、安全、网络流量和网络服务等。例如, 下载一个包含有用户界面并且经常被大量用户访问的 applet 可能需要花很长的时间, 然而如果知道该 applet 要运行在一个公司内部网的受控环境中, 那么, 在这种情况下, 该 applet 将拥有一个完全可接受的下载速度。

此外, 设计应用程序时, 还要考虑繁重的处理应当在哪里执行。例如, 如果客户程序在一个蜂窝电话或呼机中执行, 服务器应当完成尽量多的计算和数据处理, 而客户程序只应显示结果就可以了。然而, 设计在一个强大的台式机平台上运行的大型财务分析系统则应当在客户机上完成其复杂计算。应用的客户程序和 applet 用户界面通常都是用 Swing API 创建的, 该 API 可从标准版 Java 2 平台(Java 2 Platform, Standard Edition)中获得。Swing API 提供了一整套 GUI 组件(表格、树形结构、按钮等), 这些组件可以实现比用一个典型的 HTML 页面所能实现的交互性更强的功能。Swing 也支持 HTML 文本组件, 这个组件可以被用来显示来自一个服务器的响应。客户程序可以直接访问 Enterprise Bean 层或企业信息系统层。但应谨慎实现这种程序。绕过 EJB 层的程序可以使用 JDBC API 来访问一个关系型数据库, 但应被限制在对数据库表格进行维护等这样的管理任务上。

3. 设计基于 Web 的应用程序(Designing Web-Based Applications)

基于 Web 的应用程序是基于浏览器的, 如果它们运行在 Internet 上, 可能被全世界的人

访问。当设计一个基于 Web 的应用程序时，不仅需要决定用什么来处理内容和应用逻辑(HTML、XML、JSP 页面及 Servlets)，而且还应当考虑如何使该应用程序国际化。

可以在 J2EE 平台上实现的基于 Web 的应用程序有 4 种。从简单到复杂排列为：

- 基本 HTML。
- 带基本 JSP 页面或 Servlets 的 HTML。
- 带 JavaBean 类的 JSP 页面。
- 将应用逻辑根据功能划分成区域的高度结构化的应用。

当设计一个基于 Web 的应用程序时，需要决定用什么来建立它。假设是从建立一个简单的应用程序开始着手，如果认为以后会给该应用程序添加功能，那么当初的设计应当适应今后发展的需要。

Servlets 是实现动态内容的一种简便的、平台独立的、Web 服务器独立的方式。JSP 页面是开发 Servlets 的一种基于文本的、以显示为中心的方式。JSP 页面提供了 Servlets 的所有优点，并且当与一个 JavaBeans 类结合在一起时，提供了一种使内容和显示逻辑分开的简单方式。分开内容和显示逻辑的好处是，更新页面外观的人员不必懂得 Java 代码，而更新 JavaBeans 类的人员也不必是设计网页的行家里手。JSP 页面和 Servlet 都比公共网关接口(CGI)更可取，因为 CGI 不是平台独立的，使用了更多系统开销，而且没有一个简单的方法来访问参数数据，并将这些数据发送给一个程序。可以用带 JavaBeans 类的 JSP 页面来定义 Web 模板，以建立一个由具有相似外观的页面组成的网站。JavaBeans 类完成数据提供，这样在模板中就没有 Java 代码，这意味着这些模板可以由一个 HTML 编写人员来维护。

在选择使用一个 Servlet 还是一个 JSP 页面时，要注意的是，Servlets 是一个程序设计工具，它最适用于不需要频繁修改的低级应用功能；而 JSP 页面则通过以显示为中心的描述性的方法将动态内容和逻辑结合在一起。对于一个使用 JSP 页面的、简单的、基于 Web 的应用程序，可以使用定制标记或者 scriptlets(scriptlets 是直接嵌入在 JSP 页面中的很小的 Java 代码段)，而不是使用 JavaBeans 类来将内容与应用逻辑结合起来，定制标记被打包到一个标记库中，并被引入到一个 JSP 页面中。

一个国际化的基于 Web 的应用程序首先要让用户选择一种语言，然后，根据该选定语言加载应用的正文方式。对被支持的每种语言而言，应用正文都被存储在一个外部文件中，并且与另外一个文件的关键词相对应，应用代码使用这些关键词以及选定的语言来加载正确的文本。国际化 API 还提供类来根据选定的语言格式化日期和货币格式。一旦制订了使应用程序国际化的细节，就可以决定用什么来实现它了。

概括起来，一个基于 Web 的应用程序使用 HTML 来显示数据；用 XML 来定义数据以使其可被另一个程序读取并处理；使用 JSP 页面或 Servlets 来管理用户与业务层或存储层之间的数据流。

1.1.4 Java Servlet

Servlet 是位于 Web 服务器端的 Java 应用程序，与传统的从命令行启动的 Java 应用程序不同，Servlet 由 Web 服务器进行加载，该 Web 服务器必须包含支持 Servlet 的 Java 虚拟机。

Java Servlet 是 Java 中非常重要的技术之一，更是 Web 应用程序的基础，下面就介绍 Java Servlet 的相关知识。

1. 什么是 Java Servlet

Servlet 是使用 Java Servlet 应用程序设计接口(API)及相关类和方法的 Java 程序。Java Servlet API 定义了一个 Servlet 和 Java 交互的服务器之间的一个标准接口，这使得 Servlets 具有跨服务器平台的特性。除了 Java Servlet API，Servlet 还可以使用用以扩展和添加到 API 的 Java 类软件包。

Servlet 在启用 Java 的 Web 服务器上或应用服务器上运行并扩展了该服务器的能力。Java Servlet 对于 Web 服务器就好像 Java applet 对于 Web 浏览器，Servlet 的主要功能在于可以事先交互式地浏览和修改数据，生成动态 Web 内容。Servlet 装入 Web 服务器并在 Web 服务器内执行，而 applet 装入 Web 浏览器并在 Web 浏览器内执行。

2. Servlet 生命周期

Servlet 的生命周期在被装入 Web 服务器的内存时开始，并在终止或重新装入 Servlet 时结束。Servlet 在下列时刻被装入 Web 服务器的内存：

- 如果已配置自动装入选项，则在启动服务器时自动装入。
- 在服务器启动后，客户机首次向 Servlet 发出请求时。
- 重新装入 Servlet 时。

装入 Servlet 后，Web 服务器创建一个 Servlet 实例并且调用 Servlet 的 `init()` 方法。在初始化阶段，Servlet 初始化参数被传递给 Servlet 配置对象，请求处理对于到达服务器的客户机请求，服务器创建对应于请求的一个“请求”对象和一个“响应”对象。然后服务器调用 Servlet 的 `service()` 方法，该方法用于传递“请求”和“响应”对象。`service()` 方法从“请求”对象获得请求信息，处理该请求后用“响应”对象的方法将响应传回客户机。

当服务器不再需要 Servlet，或重新装入 Servlet 的新实例时，服务器会调用 Servlet 的 `destroy()` 方法，来结束这个 Servlet。对于更多的客户端请求，Server 创建新的请求和响应对象，仍然激活此 Servlet 的 `service()` 方法，将这两个对象作为参数传递给它。如此重复以上的循环，但无需再次调用 `init()` 方法，所以一般情况下 Servlet 只初始化一次。

3. Java Servlet 的优势

Servlet 提供了 Java 应用程序的所有优势——可移植、稳健、易开发。除此之外，Java Servlet 还具有以下几个方面的优势：

- Servlet 可以和其他资源(文件、数据库、Applet、Java 应用程序等)交互，以生成返回给客户端的响应内容，如果需要，还可以保存请求—响应过程中的信息。
- 采用 Servlet，服务器可以完全授权对本地资源的访问(如数据库)，并且 Servlet 自身将会控制外部用户的访问数量及访问性质。
- Servlet 可以是其他服务的客户端程序。例如，它们可以用于分布式的应用系统中，可以从本地硬盘，或者通过网络从远端硬盘激活 Servlet。

- Servlet 可被链接。一个 Servlet 可以调用另一个或一系列 Servlet，即成为它的客户端。
- 采用 ServletTag 技术，可以在 HTML 页面中动态调用 Servlet。
- Servlet API 与协议无关，它并不对传递它的协议有任何假设。

但是，Java Servlet 也不是没有缺点。和传统的 CGI、ISAPI、NSAPI 一样，Java Servlet 是利用输出 HTML 语句来实现动态网页的，这就使得 Java Servlet 不利于用来开发整个网站，于是 SUN 公司又推出了 JSP(Java Server Pages)。

1.1.5 Struts

Struts 就是在 JSP Model 2 的基础上实现了 MVC(模型—视图—控制器)设计模式的 Web 框架。基于 Struts 构架的 Web 应用程序基本上符合 JSP Model 2 的设计标准，可以说是 MVC 设计模式的一种变化类型。

Struts 由一组相互协作的类、Servlet 以及标签库和独立于框架工作的验证类组成。Struts 是基于 MVC 实现的，拥有自己的控制器、模型、视图。通过模型层，Struts 可以方便地与 EJB、JDBC、ORB 等数据库访问技术相结合；通过视图层，可以与 JSP、Velocity Templates、XSL 等组件相结合。Struts 的工作原理如图 1-3 所示。

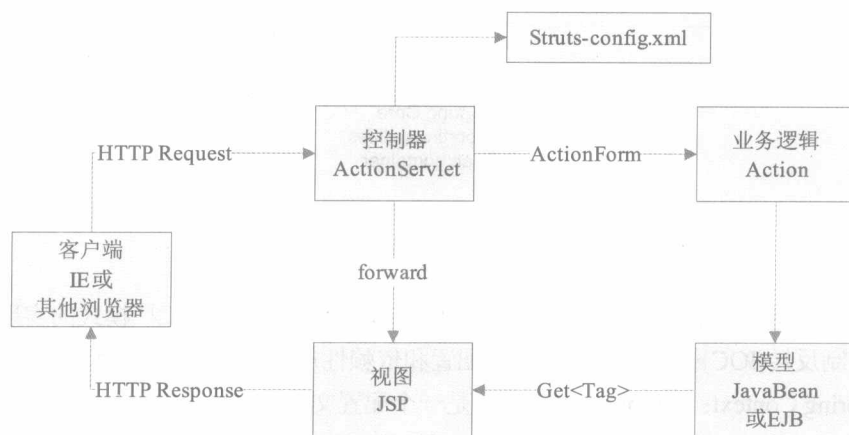


图 1-3 Struts 工作原理

- 视图：视图主要由 JSP 建立，在 Struts 中，JSP 可以使用扩展自定义标签库(TagLib)来创建完全国际化的用户界面。目前的标签库包括 Bean Tags、HTML tags、Logic Tags、Nested Tags 以及 Template Tags 等。
- 模型：模型主要表示一个系统的状态，在 Struts 的体系结构中，系统的状态主要由 ActionForm Bean 体现，一般情况下，这些状态是非持久性的。
- 控制器：在 Struts 中，基本的控制器组件是 ActionServlet 类中的实例 Servlet，实际使用的 Servlet 在配置文件中由一组映射(由 ActionMapping 类来描述)进行定义。对于业务逻辑的操作则主要由 Action、ActionMapping、ActionForward 这几个组件协调完成，其中 Action 扮演了真正的业务逻辑的实现者，ActionMapping 与 ActionForward 则指