



高等院校规划教材

主 编 王 红  
副主编 高 玲 叶灵君

# Java Web应用开发技术实用教程 习题解答与实验指导



强调理论与实践相结合，注重专业技术技能的培养  
引入典型工程案例，提高工程实用技术的能力



中国水利水电出版社  
www.waterpub.com.cn

21 世纪高等院校规划教材

# Java Web 应用开发技术实用教程

## 习题解答与实验指导

主 编 王 红

副主编 高 玲 叶灵君

ISBN 978-7-208-21271-4	21 世纪高等院校规划教材	中国版本图书馆 CIP 数据核字 (2008) 第 106692 号	2008 年 8 月第 1 版 2008 年 8 月第 1 次印刷 0001—4000 册	28.00 元
北京航空航天大学出版社	北京航空航天大学出版社	北京航空航天大学出版社	北京航空航天大学出版社	北京航空航天大学出版社
地址: 北京市海淀区学院路 5 号	地址: 北京市海淀区学院路 5 号	地址: 北京市海淀区学院路 5 号	地址: 北京市海淀区学院路 5 号	地址: 北京市海淀区学院路 5 号
电话: (010) 62522266	电话: (010) 62522266	电话: (010) 62522266	电话: (010) 62522266	电话: (010) 62522266
网址: www.waterpub.com.cn	网址: www.waterpub.com.cn	网址: www.waterpub.com.cn	网址: www.waterpub.com.cn	网址: www.waterpub.com.cn
E-mail: mcshamel@263.net	E-mail: mcshamel@263.net	E-mail: mcshamel@263.net	E-mail: mcshamel@263.net	E-mail: mcshamel@263.net
sales@waterpub.com.cn	sales@waterpub.com.cn	sales@waterpub.com.cn	sales@waterpub.com.cn	sales@waterpub.com.cn
发行	发行	发行	发行	发行
出版	出版	出版	出版	出版
作 者	作 者	作 者	作 者	作 者
主 编 王 红	主 编 王 红	主 编 王 红	主 编 王 红	主 编 王 红
副主编 高 玲 叶灵君	副主编 高 玲 叶灵君	副主编 高 玲 叶灵君	副主编 高 玲 叶灵君	副主编 高 玲 叶灵君
公 司	公 司	公 司	公 司	公 司
北京航空航天大学出版社	北京航空航天大学出版社	北京航空航天大学出版社	北京航空航天大学出版社	北京航空航天大学出版社



## 内 容 提 要

本书是《Java Web 应用开发技术实用教程》(王红主编,中国水利水电出版社 2008 年出版)的配套习题解答与实验指导用书。全书分为四大部分:第一部分是主教材每章内容的重点和难点;第二部分是主教材各章的习题解答;第三部分是各章的实验指导;第四部分是综合测试题,包括 4 套模拟题。这些内容对于巩固和深入理解教材的内容,提高独立思考、独立分析问题的能力,都是十分有益的。

本书适合作为各类高等院校、高职高专计算机类专业和各种培训班的教材,也可供广大科技人员和感兴趣的读者参考。

本书所有程序全部运行通过,其源代码可以在中国水利水电出版社网站下载,网址为 <http://www.waterpub.com.cn/softdown/>。

## 图书在版编目(CIP)数据

Java Web 应用开发技术实用教程习题解答与实验指导 /  
王红主编. —北京:中国水利水电出版社, 2008

21 世纪高等院校规划教材

ISBN 978-7-5084-5757-4

I. J… II. 王… III. JAVA 语言—程序设计—高等学校—  
教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 106697 号

书 名	Java Web 应用开发技术实用教程习题解答与实验指导
作 者	主 编 王 红 副主编 高 玲 叶灵君
出版 发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 63202266 (总机)、68367658 (营销中心)、82562819 (万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	184mm×260mm 16 开本 15.25 印张 373 千字
版 次	2008 年 8 月第 1 版 2008 年 8 月第 1 次印刷
印 数	0001—4000 册
定 价	25.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

## 序

随着计算机科学与技术的飞速发展,计算机的应用已经渗透到国民经济与人们生活的各个角落,正在日益改变着传统的人类工作方式和生活方式。在我国高等教育逐步实现大众化后,越来越多的高等院校会面向国民经济发展的第一线,为行业、企业培养各级各类高级应用型专门人才。为了大力推广计算机应用技术,更好地适应当前我国高等教育的跨越式发展,满足我国高等院校从精英教育向大众化教育的转变,符合社会对高等院校应用型人才培养的各类要求,我们成立了“21世纪高等院校规划教材编委会”,在明确了高等院校应用型人才培养模式、培养目标、教学内容和课程体系的框架下,组织编写了本套“21世纪高等院校规划教材”。

众所周知,教材建设作为保证和提高教学质量的重要支柱及基础,作为体现教学内容和教学方法的知识载体,在当前培养应用型人才中的作用是显而易见的。探索和建设适应新世纪我国高等院校应用型人才体系需要的配套教材已经成为当前我国高等院校教学改革和教材建设工作面临的紧迫任务。因此,编委会经过大量的前期调研和策划,在广泛了解各高等院校的教学现状、市场需求,探讨课程设置、研究课程体系的基础上,组织一批具备较高的学术水平、丰富的教学经验、较强的工程实践能力的学术带头人、科研人员和主要从事该课程教学的骨干教师编写出一批有特色、适用性强的计算机类公共基础课、技术基础课、专业及应用技术课的教材以及相应的教学辅导书,以满足目前高等院校应用型人才的需要。本套教材消化和吸收了多年来已有的应用型人才探索与实践成果,紧密结合经济全球化时代高等院校应用型人才工作的实际需要,努力实践,大胆创新。教材编写采用整体规划、分步实施、滚动立项的方式,分期分批地启动编写计划,编写大纲的确定以及教材风格的定位均经过编委会多次认真讨论,以确保该套教材的高质量和实用性。

教材编委会分析研究了应用型人才与研究型人才在培养目标、课程体系和内容编排上的区别,分别提出了3个层面上的要求:在专业基础类课程层面上,既要保持学科体系的完整性,使学生打下较为扎实的专业基础,为后续课程的学习做好铺垫,更要突出应用特色,理论联系实际,并与工程实践相结合,适当压缩过多过深的公式推导与原理性分析,兼顾考研学生的需要,以原理和公式结论的应用为突破口,注重它们的应用环境和方法;在程序设计类课程层面上,把握程序设计方法和思路,注重程序设计实践训练,引入典型的程序设计案例,将程序设计类课程的学习融入案例的研究和解决过程中,以学生实际编程解决问题的能力为突破口,注重程序设计的实现;在专业技术应用层面上,积极引入工程案例,以培养学生解决工程实际问题的能力为突破口,加大实践教学内容的比重,增加新技术、新知识、新工艺的内容。

本套规划教材的编写原则是:

在编写中重视基础,循序渐进,内容精炼,重点突出,融入学科方法论内容和科学理念,反映计算机技术发展要求,倡导理论联系实际和科学的思想方法,体现一级学科知识组织的层次结构。主要表现在:以计算机学科的科学体系为依托,明确目标定位,分类组织实施,兼容互补;理论与实践并重,强调理论与实践相结合,突出学科发展特点,体现

学科发展的内在规律；教材内容循序渐进，保证学术深度，减少知识重复，前后相互呼应，内容编排合理，整体结构完整；采取自顶向下设计方法，内涵发展优先，突出学科方法论，强调知识体系可扩展的原则。

本套规划教材的主要特点是：

(1) 面向应用型高等院校，在保证学科体系完整的基础上不过度强调理论的深度和难度，注重应用型人才的专业技能和工程实用技术的培养。在课程体系方面打破传统的研究型人才培养体系，根据社会经济发展对行业、企业的工程技术需要，建立新的课程体系，并在教材中反映出来。

(2) 教材的理论知识包括了高等院校学生必须具备的科学、工程、技术等方面的要求，知识点不要求大而全，但一定要讲透，使学生真正掌握。同时注重理论知识与实践相结合，使学生通过实践深化对理论的理解，学会并掌握理论方法的实际运用。

(3) 在教材中加大能力训练部分的比重，使学生比较熟练地应用计算机知识和技术解决实际问题，既注重培养学生分析问题的能力，也注重培养学生思考问题、解决问题的能力。

(4) 教材采用“任务驱动”的编写方式，以实际问题引出相关原理和概念，在讲述实例的过程中将本章的知识点融入，通过分析归纳，介绍解决工程实际问题的思想和方法，然后进行概括总结，使教材内容层次清晰，脉络分明，可读性、可操作性强。同时，引入案例教学和启发式教学方法，便于激发学习兴趣。

(5) 教材在内容编排上，力求由浅入深，循序渐进，举一反三，突出重点，通俗易懂。采用模块化结构，兼顾不同层次的需求，在具体授课时可根据各校的教学计划在内容上适当加以取舍。此外还注重了配套教材的编写，如课程学习辅导、实验指导、综合实训、课程设计指导等，注重多媒体的教学方式以及配套课件的制作。

(6) 大部分教材配有电子教案，以使教材向多元化、多媒体化发展，满足广大教师进行多媒体教学的需要。电子教案用 PowerPoint 制作，教师可根据授课情况任意修改。相关教案的具体情况请到中国水利水电出版社网站 [www.waterpub.com.cn](http://www.waterpub.com.cn) 下载。此外还提供相关教材中所有程序的源代码，方便教师直接切换到系统环境中教学，提高教学效果。

总之，本套规划教材凝聚了众多长期在教学、科研一线工作的教师及科研人员的教学科研经验和智慧，内容新颖，结构完整，概念清晰，深入浅出，通俗易懂，可读性、可操作性和实用性强。本套规划教材适用于应用型高等院校各专业，也可作为本科院校举办的应用技术专业的课程教材，此外还可作为职业技术学院和民办高校、成人教育的教材以及从事工程应用的技术人员的自学参考资料。

我们感谢该套规划教材的各位作者为教材的出版所做出的贡献，也感谢中国水利水电出版社为选题、立项、编审所做出的努力。我们相信，随着我国高等教育的不断发展和高校教学改革的不深入，具有示范性并适应应用型人才培养的精品课程教材必将进一步促进我国高等院校教学质量的提高。

我们期待广大读者对本套规划教材提出宝贵意见，以便进一步修订，使该套规划教材不断完善。

21 世纪高等院校规划教材编委会

2004 年 8 月

# 前 言

本书是《Java Web 应用开发技术实用教程》(王红主编,中国水利水电出版社 2008 年出版)的配套教材。为了配合教学,提高学生对知识点的理解并提高学生的操作能力,我们精心编写了本书。全书分为四大部分:第一部分是主教材每章内容的重点和难点;第二部分是主教材各章的习题解答;第三部分是各章的实验指导;第四部分是综合测试题(共 4 套模拟题)。这些内容对于巩固和深入理解教材的内容,提高独立思考、独立分析问题的能力都是十分有益的。

本书在内容安排上与主教材配套,重点和难点、实验指导和习题分别按主教材的各章顺序编排。主教材中的内容非常丰富全面,习题和实验题很广,有的习题和实验有相当难度。本书习题解答部分对主教材中的全部习题做了详尽解答,便于读者学习和参考。实验指导主要包括实验内容和操作步骤,其中包含大量的十分有用的程序,而且所有程序均调试通过。每个实验都提供了多套实验操作题目,有利于提高学生的操作及应用能力。第 5 章和第 9 章中还提供了综合实验,学生可以通过所学知识发挥创意设计一个综合作业,以展示自己的应用能力和创意水平。4 套模拟题均附有参考答案,有利于读者练习提高,本书可以作为高等院校该课程期末考试命题参考,也可作为学生综合检验学习效果的自测题。

本书的基本结构为:第 1 章 JSP 概述,第 2 章 JSP 基础,第 3 章 JSP 隐含对象,第 4 章 JSP 中使用的数据库,第 5 章 JSP 与 JavaBean,第 6 章 EL 表达式语言,第 7 章 标准标记——JSTL,第 8 章 Servlet 技术,第 9 章 Struts 架构。

本书适合作为各类高等院校、高职高专计算机类专业和各种培训班的教材,也可供广大科技人员和感兴趣的读者参考。

本书由王红教授任主编,高玲、叶灵君任副主编。高玲老师和叶灵君老师参与了本书大纲的讨论和编写工作。王晓卉、蔡媛媛、李晶晶、惠伟、郑雷雷、李健波、徐娜、黄雯、魏秋红参与了编写初稿和校稿等工作。本书在编写过程中得到了许多支持和帮助,在此表示衷心感谢。最后,感谢中国水利水电出版社对本书出版的支持与协助。

限于作者水平,书中难免有错误和不足之处,恳请读者批评指正,作者将不胜感激。在阅读本书时,如果发现任何问题请发 E-mail 至 wanghong106@163.com 提出宝贵意见,在此一并表示感谢。

王 红

2008 年 6 月

# 目 录

序

前言

第 1 章 JSP 概述	1
1.1 重点	1
1.2 难点	2
1.3 习题解答	2
实验 1 搭建 JSP 运行的实验环境	3
一、实验目的	3
二、实验工具	3
三、实验步骤	3
四、实验练习	5
第 2 章 JSP 基础	7
2.1 重点	7
2.2 难点	9
2.3 习题解答	10
实验 2 JSP 脚本、指令和动作	12
一、实验目的	12
二、实验工具	12
三、实验步骤	12
四、实验练习	14
第 3 章 JSP 隐含对象	19
3.1 重点	19
3.2 难点	19
3.3 习题解答	20
实验 3 JSP 隐含对象的使用	24
一、实验目的	24
二、实验工具	24
三、实验步骤	24
四、实验练习	35
第 4 章 JSP 中使用数据库	38
4.1 重点	38
4.2 难点	38
4.3 习题解答	40
实验 4 JDBC 技术	52

一、实验目的 .....	52
二、实验工具 .....	52
三、实验步骤 .....	52
四、实验练习 .....	55
<b>第 5 章 JSP 与 JavaBean .....</b>	<b>73</b>
5.1 重点 .....	73
5.2 难点 .....	74
5.3 习题解答 .....	75
实验 5 JSP 中使用 Java Bean .....	79
一、实验目的 .....	79
二、实验工具 .....	79
三、实验步骤 .....	79
四、实验练习 .....	121
<b>第 6 章 表达式语言 .....</b>	<b>131</b>
6.1 重点 .....	131
6.2 难点 .....	131
6.3 习题解答 .....	132
实验 6 JSP 中使用 EL 表达式 .....	132
一、实验目的 .....	132
二、实验工具 .....	132
三、实验步骤 .....	133
四、实验练习 .....	137
<b>第 7 章 标准标记——JSTL .....</b>	<b>140</b>
7.1 重点 .....	140
7.2 难点 .....	141
7.3 习题解答 .....	141
实验 7 JSP 中使用 JSTL .....	142
一、实验目的 .....	142
二、实验工具 .....	142
三、实验步骤 .....	142
四、实验练习 .....	149
<b>第 8 章 Servlet 技术 .....</b>	<b>152</b>
8.1 重点 .....	152
8.2 难点 .....	153
8.3 习题解答 .....	154
实验 8 Servlet 与过滤器 .....	154
一、实验目的 .....	154
二、实验工具 .....	154
三、实验步骤 .....	154



四、实验练习 .....	161
<b>第 9 章 Struts 架构 .....</b>	<b>169</b>
9.1 重点 .....	169
9.2 难点 .....	171
9.3 习题解答 .....	171
实验 9 Struts 架构 .....	171
一、实验目的 .....	171
二、实验工具 .....	172
三、实验步骤 .....	172
四、实验练习 .....	190
模拟试题 1 .....	200
模拟试题 1 答案 .....	205
模拟试题 2 .....	209
模拟试题 2 答案 .....	215
模拟试题 3 .....	218
模拟试题 3 答案 .....	223
模拟试题 4 .....	226
模拟试题 4 答案 .....	232

# 第 1 章 JSP 概述

## 1.1 重点

本章重点是理解 JSP 的运行原理。JSP 的实现是基于 Servlet 的，JSP 页面在运行之前要被翻译成 Java Servlet。当 JSP 容器接到对一个 JSP 页面的请求后，首先判断与 JSP 文件对应的 Servlet 类是否存在或比 JSP 文件老（意味着 JSP 源文件自从上次被翻译后已经改变），如果是，容器就会重新创建一个等价的 Servlet 类并编译它。如果 Servlet 类存在并且是最新版只是其实例并未运行，容器就载入该 Servlet 类并创建一个实例。最后，容器启动一个线程在载入的实例中处理当前请求。图 1-1 给出 JSP 的详细工作过程。

本章重点是理解 JSP 的运行原理。

JSP 的实现是基于 Servlet 的，JSP 页面在运行之前要被翻译成 Java Servlet。当 JSP 容器接到对一个 JSP 页面的请求后，首先判断与 JSP 文件对应的 Servlet 类是否存在或比 JSP 文件老（意味着 JSP 源文件自从上次被翻译后已经改变），如果是，容器就会重新创建一个等价的 Servlet 类并编译它。如果 Servlet 类存在并且是最新版只是其实例并未运行，容器就载入该 Servlet 类并创建一个实例。最后，容器启动一个线程在载入的实例中处理当前请求。图 1-1 给出 JSP 的详细工作过程。

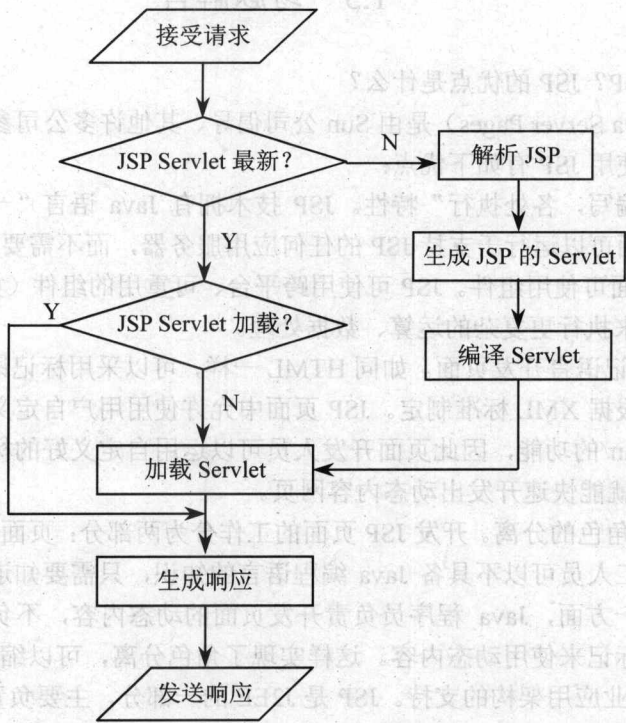


图 1-1 JSP 与 Servlet 关系示意图

因为 JSP 在执行第一次前会被编译成 Servlet 的类文件，再重复调用执行时，就直接执行第一次产生的 Servlet，而不用再重新把 JSP 翻译成 Servlet。因此，除了第一次的编译多花一点时间外，之后 JSP 和 Servlet 的执行速度几乎相同。

## 1.2 难点

本章难点是安装 JSP 运行环境。

安装 JSP 运行环境主要经过以下步骤：安装 J2SDK、安装 Tomcat 以及设置环境变量（具体步骤参见主教材）。J2SDK 是 Java 虚拟机，Tomcat 是 Web 服务器。其中，设置环境变量为可选，与 Tomcat 的版本有关。当测试 Tomcat 主页或提供的 JSP 服务不成功时，再考虑设置环境变量。环境变量共有 4 个：path、classpath、java\_home、tomcat\_home。

变量名：path，变量值：H:\Program Files\Java\jdk1.6.0\_01\bin，其中 H:\Program Files\Java\jdk1.6.0\_01 是 J2SDK 的安装目录。

变量名：classpath，变量值：H:\Program Files\Java\jdk1.6.0\_01\lib\rt.jar;.; H:\Program Files\Java\jdk1.6.0\_01\lib\tools.jar。

变量名：java\_home，变量值：J2SDK 的安装目录。

变量名：tomcat\_home，变量值：Tomcat 的安装目录。

## 1.3 习题解答

1. 什么是 JSP? JSP 的优点是什么?

答：JSP (Java Server Pages) 是由 Sun 公司倡导、其他许多公司参与一起建立的一种动态网页技术标准。使用 JSP 有如下优点：

(1) “一次编写，各处执行”特性。JSP 技术拥有 Java 语言“一次编写，各处执行”的特点，即 JSP 页面可以运行于支持 JSP 的任何应用服务器，而不需要对代码做任何修改。

(2) JSP 页面可使用组件。JSP 可使用跨平台、可重用的组件（如 JavaBean 或 Enterprise JavaBean 组件）来执行更复杂的运算、数据处理。

(3) 采用标记语言开发页面。如同 HTML 一样，可以采用标记语言来开发 JSP 页面，而且标记语言完全根据 XML 标准制定。JSP 页面中允许使用用户自定义标记，用户自定义标记可以完成 JavaBean 的功能，因此页面开发人员可以运用自定义好的标记，无须在页面中再写复杂的 Java 代码就能快速开发出动态内容网页。

(4) 实现了角色的分离。开发 JSP 页面的工作分为两部分：页面美工和页面的动态内容。一方面，页面美工人员可以不具备 Java 编程语言的知识，只需要知道标记如何使用和页面布局如何漂亮。另一方面，Java 程序员负责开发页面的动态内容，不负责页面美工，然后由页面美工人员通过标记来使用动态内容。这样实现了角色分离，可以缩短应用开发周期。

(5) N 层企业应用架构的支持。JSP 是 J2EE 的一部分，主要负责 Web 服务器前端显示，后端主要依赖 EJB 和 Java Bean 构建而成。它们与客户端和数据库服务器端一起构成 B/S/S 三层或三层以上的企业应用架构。

2. 简述 JSP 的执行过程。

答：JSP 的执行过程主要包括以下步骤：

(1) 客户端发出 Request 请求。

(2) JSP 容器将 JSP 翻译成 Servlet 的源代码。

(3) 将产生的 Servlet 源代码编译成 CLASS 文件，并加载到内存执行。

(4) 把 Response 响应输出至客户端。

### 3. JSP 和 Servlet 的关系如何?

答: JSP 和 Servlet 之间有着内在的“血缘”关系。首先, JSP 的实现是基于 Servlet 的, JSP 页面在运行之前要被翻译成 Servlet, 然后由 Servlet 处理 HTTP 请求并生成响应信息。JSP 充分利用了 Servlet 的平台独立性、安全性、可升级性以及强壮性等优点。另外, JSP 页面和 Servlet 程序可以相互操作, JSP 页面可以包含从 Servlet 程序输出的内容, 可以将内容输出到 Servlet 程序。反过来 Servlet 程序也可以包含从 JSP 页面输出的内容并且可以将内容输出到 JSP 页面中。

JSP 比 Servlet 具有更多优点。首先, JSP 页面使用标记, 省略掉传统 Java 代码中的 import、print 和 try...catch 等语句, 可以使用任何流行的 HTML 创作工具来创建 JSP 页面。其次, JSP 页面由服务器自动编译, JSP 容器会自动处理大量 Servlet 编码工作, 如获得 Session 等隐含对象, 使用起来更为方便。

在构建 Web 应用时, Servlet 和 JSP 结合使用。

### 4. 什么是 JSP 容器? 你知道的 JSP 容器有哪些?

答: JSP 容器是由许多 Servlet 构成的程序实体, 它将请求信息传递给 JSP 页面并解释、执行 JSP 页面。

5. B 6. C 7. A 8. A

## 实验 1 搭建 JSP 运行的实验环境

### 一、实验目的

参见主教材。

### 二、实验工具

参见主教材。

### 三、实验步骤

1. 按书中演示安装 JDK。

答: 见主教材图 1-5 至图 1-8。

2. 按书中演示安装 Tomcat。

答: 见主教材图 1-10 至图 1-16。

3. 按书中演示配置环境变量。

答: 见主教材图 1-9。

4. 启动 Tomcat 服务器, 观察主页内容。

答: 见主教材图 1-7 至图 1-20。

5. 观察 Tomcat 安装目录下的各个子目录内容。

答: 见主教材图 1-17。

6. 将 exa1\_1.jsp 拷贝到 Tomcat 安装目录的 webapps 目录下的 ROOT 目录下, 运行该页面, 观察其对应的 Servlet 源代码。

答: 源代码在 Tomcat 的安装目录\work\Standalone\localhost 目录下找到。源代码如下:

```
package org.apache.jsp;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
public final class exa1_005f1_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {
    private static java.util.List _jspx_dependants;
    public Object getDependants() {
        return _jspx_dependants;
    }
    public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {
        JspFactory _jspxFactory = null;
        PageContext pageContext = null;
        HttpSession session = null;
        ServletContext application = null;
        ServletConfig config = null;
        JspWriter out = null;
        Object page = this;
        JspWriter _jspx_out = null;
        PageContext _jspx_page_context = null;
        try {
            _jspxFactory = JspFactory.getDefaultFactory();
            response.setContentType("text/html;charset=gb2312");
            pageContext = _jspxFactory.getPageContext(this, request, response,
                null, true, 8192, true);
            _jspx_page_context = pageContext;
            application = pageContext.getServletContext();
            config = pageContext.getServletConfig();
            session = pageContext.getSession();
            out = pageContext.getOut();
            _jspx_out = out;
            out.write("\r\n");
            out.write("<HTML>\r\n");
            out.write("<BODY bgcolor=cyan>\r\n");
            out.write("<FONT size=4>\r\n");
            out.write("<P> 我的第一个 jsp 页面。你好吗? 朋友 \r\n");
            int sum=100;
            out.write("\r\n");
            out.write("<BR>\r\n");
            out.println("sum="+sum);
        }
    }
}
```

```

out.write("\r\n");
out.write("</FONT>\r\n");
out.write("</BODY>\r\n");
out.write("</HTML>\r\n");
out.write("\r\n");
} catch (Throwable t) {
    if (!(t instanceof SkipPageException)){
        out = _jspx_out;
        if (out != null && out.getBufferSize() != 0)
            out.clearBuffer();
        if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
    }
} finally {
    if (_jspxFactory != null) _jspxFactory.releasePageContext(_jspx_page_context);
}
}
}
}

```

7. 将 exa1\_1.jsp 拷贝到 Tomcat 安装目录的 webapps 目录下的 examples 目录下，运行该页面，观察其对应的 Servlet 源代码。

答：源代码在 Tomcat 的安装目录\work\Standalone\localhost\examples 目录下找到。源代码同上。

8. 将 exa1\_1.jsp 拷贝到 Tomcat 安装目录的 webapps 目录下的 webdav 目录下，运行该页面，观察其对应的 Servlet 源代码。

答：源代码在 Tomcat 的安装目录\work\Standalone\localhost\webdav 目录下找到。源代码同上。

#### 四、实验练习

编写一个 JSP 页面，用于显示 1~100 的整数和。

答：源代码如下：

```

<%@ page contentType="text/html;charset=GB2312"%>
<HTML>
<BODY BGCOLOR=cyan>
<FONT Size=4>
<%
int i,sum=0;
for (i=1;i<=100;i++)
{
    sum=sum+i;
}
%>
<P> 1 到 100 的整数和:
<BR>
<%=sum%>

```

```

</FONT>
</BODY>
</HTML>

```

运行结果如图 1-2 所示。

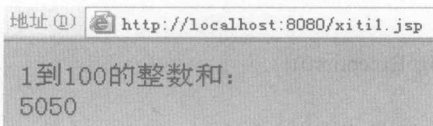


图 1-2 实验练习题运行结果

```

        out.write("\n");
        out.write("<FONT>\n");
        out.write("<BODY>\n");
        out.write("<HTML>\n");
        out.write("\n");
        catch (Throwable t) {
            if (!instanceOf Skip
                out = jsp_out;
            if (out != null) && out.isError()
                out.clearBuffer();
            if (jsp_page_context != null) jsp_page_context.handlePageError(t);
        } finally {
            if (jsp_session != null) jsp_session.releasePageContext(jsp_page_context);
        }
    }

    7. 将 xiti1.jsp 拷贝到 Tomcat 安装目录的 webapps 目录下，运行该
    页面，观察其对应的 Servlet 源代码。
    答：源代码在 Tomcat 的安装目录 work\standalone\localhost\examples 目录下找到。源代码
    同上。

    8. 将 xiti1.jsp 拷贝到 Tomcat 安装目录的 webapps 目录下，运行该页
    面，观察其对应的 Servlet 源代码。
    答：源代码在 Tomcat 的安装目录 work\standalone\localhost\webapps 目录下找到。源代码
    同上。

    四、实验练习

    编写一个 JSP 页面，用于显示 1-100 的整数和。
    答：源代码如下：
    <@page contentType="text/html; charset=UTF-8" %>
    <HTML>
    <BODY BGCOLOR=#cyan>
    <FONT size=4>
    <P>
    int sum=0;
    for (i=1;i<=100;i++)
    sum=sum+i;
    </P>
    <BR>
    <P>1 到 100 的整数和:
    </P>
    </BODY>
    </HTML>

```

## 第 2 章 JSP 基础

### 2.1 重点

本章重点是掌握 JSP 页面的基本组成，掌握每个 JSP 标记的语法和特点。

JSP 页面是由传统的 HTML 页面标记加上 JSP 标记组成的，JSP 标记具体包括以下 4 种标记：注释、指令、脚本和动作。

#### 1. 注释。

为了提高程序代码的可读性和代码重用，编写代码时应在适当位置给予必要的注释。在 JSP 页面中可加入三种类型的注释：HTML 注释、JSP 注释和脚本注释。

(1) HTML 注释的语法形式为：`<!-- 注释 -->`，其特点是：它们被不做任何改动地加入到 JSP 响应中，发送给浏览器，由浏览器负责忽略。因此客户通过浏览器查看源文件时可以看到 HTML 注释。另外，在 JSP 页面对应的 Servlet 中也可以看到该类注释，但它们是作为 `out.print()` 的参数出现的。

(2) JSP 注释也称为隐藏注释。其语法形式为：`<%-- 注释 --%>`，其特点是：该类注释只能在 JSP 源文件看到。在 JSP 对应的 Servlet 中看不到，在客户端也看不到，因为此类注释在翻译成 Servlet 时已经被 JSP 容器忽略。

(3) 脚本注释是在 JSP 页面的 Java 代码段中使用的注释。

其语法形式为：`// 单行注释`

或者为：`/* 多行注释 */`

其特点是：该类注释不仅能在 JSP 源文件中看到，而且能在 JSP 翻译成的 Servlet 中看到。但是在客户端看不到，因为 Servlet 运行在服务器端，运行时该类注释被忽略掉。

#### 2. 指令。

JSP 指令是针对 JSP 容器的，它将一些特殊处理信息传送到 JSP 容器，告诉容器如何处理 JSP 页面。指令主要包括 `page` 指令、`include` 指令和 `taglib` 指令。它们的语法形式可以统一为：

`<%@ 指令名 属性名 1="属性值 1" 属性名 2="属性值 2"...属性名 n="属性值 n"%>`

对应的 XML 语法形式为：

`<jsp:directive.指令名 属性名 1="属性值 1" 属性名 2="属性值 2"...属性名 n="属性值 n"/>`

其特点是：指令不会直接产生任何看得见的输出。

(1) `page` 指令用来定义和操作许多重要的页面属性，这些属性影响整个 JSP 文件（整个 Servlet 类文件）。在写 JSP 页面时，`page` 指令的许多属性可以不去理会，因为常用到它们的默认值。在页面中经常要写的属性是 `contentType`，它的默认值为“`text/html; charset=ISO-8859-1`”，我们要改成“`text/html; charset=GBK`”或“`text/html; charset=GB2312`”。

`page` 指令可以在 JSP 页面的任何地方、以任何顺序出现，而且一个页面可以包含任意数量的 `page` 指令。但是任何的属性/值对只能出现一次（除 `import` 外）。几乎在所有 JSP 页面顶



部都会看到 page 指令, 尽管不是必需的。但是, 为了程序的可读性及良好的编程习惯, 建议把它放在 JSP 页面的顶部。

(2) include 指令的作用是在 JSP 页面中该指令出现的位置静态插入一个文件。这里的静态是指当前的 JSP 页面和被嵌入的文件合并成一个新的 JSP 页面, 然后 JSP 容器将这个新页面转换成 Servlet。这里的文件可以是 HTML 页面、文本文件或者 JSP 页面, 并且这些文件应该和当前 JSP 页面在同一个 Web 应用中。

include 指令只有一个 file 属性。另外, 在一个 JSP 页面中 include 指令的数量不限。

(3) 在 JSP 页面中要使用用户自定义标记时, 需要使用 taglib 指令, 将标记库导入到 JSP 页面中。taglib 指令包含两个属性: uri 和 prefix。其中 uri 是用户自定义标记所在的标记库描述文件的 URI 地址, prefix 是标记库描述文件的前缀。具体使用见难点分析。

### 3. 脚本。

JSP 脚本用于在 JSP 页面中插入 Java 语言的成分, 可以细分为声明、代码段和表达式。声明用于定义特定于 JSP 页面 (Servlet 类) 的变量、方法和类; 代码段是嵌入 JSP 页面的 Java 语句; 表达式是 Java 输出语句的简化形式。

(1) 声明。声明有 3 个用途: 声明变量、函数或类, 让页面的其余部分能够使用。声明的变量和方法是该页面所对应的 Servlet 类的成员变量和成员方法, 声明的类是 Servlet 类的内部类。

声明的语法形式为: `<%! 声明 %>`。

等价的 XML 语法形式为: `<jsp:declaration> 声明 </jsp:declaration>`

注意: 声明不会产生任何输出。

声明中的变量等价于 Java 中的静态变量, 若同时有 n 个用户在执行此 JSP 网页时, 将会共享此变量, 一个客户的操作会影响另外的客户。因为 Servlet 默认为多线程的, 所以在对变量写操作时应该互斥访问。变量声明的最实际用途是针对静态 final 常量。由于声明的变量是 Servlet 类中的成员变量, 因此它的初始化在 JSP 页面初始化时完成, 并且初始化只做一次。

声明的方法是 JSP 页面翻译成 Servlet 类的成员方法。要注意, 因为生成的代码不在 `_jspService()` 方法中, 所以它没有权限访问隐含对象 (request、response、out 等)。如果要使用它们, 必须将之显式传递到方法中。

声明类给出了编写内部类的简便方式。代码段在访问内部类时, 必须创建该类的对象。

(2) 代码段。代码段的作用是在 JSP 页面中加入 Java 程序片段, 其中包含一个或多个有效而完整的 Java 语句。在 JSP 页面翻译成 Servlet 时, 这些语句按照它们出现的顺序罗列在 `_jspService()` 方法中。当 Web 服务器接受客户端请求时, 这些程序代码会被执行并向客户端产生输出。

其语法形式为: `<% Java 语句 %>`。

等价的 XML 语法形式为: `<jsp:scriptlet> Java 语句 </jsp:scriptlet>`。

(3) 表达式是 Java 代码段的一种简单书写方式, 用于输出表达式的结果。

表达式的语法形式为: `<%=表达式%>`。

或: `<jsp:expression> 表达式 </expression>`。

### 4. 动作。

JSP 动作标记是 JSP 页面中使用的一种特殊标记, 它利用 XML 语法格式来控制 JSP 容器