高职高专计算机教学改革新体系规划教材

JSP开发应用项目教程

张君华 主编

清华大学出版社

高职高专计算机教学改革新体系规划教材

JSP开发应用项目教程

张君华 主编

清华大学出版社
北京

内 容 简 介

本书是基于一个完整项目开展JSP开发技术教学的教材。书中围绕开发一个社区居家养老服务信息系统，通过设置系统开发准备、系统首页设计、管理员首页与系统管理设计、老年客户管理、社工信息处理、工单信息处理、信息查询与统计、信息发布与维护等章节，将对JSP程序设计技术的介绍与实际系统的开发融为一体，通过“做中学”学习JSP技术。

本书要求学习者有一定的Java程序设计基础和基于Dreamweaver CS的静态网页制作基础。本书可作为高职高专院校、各类培训机构的JSP学习教材，也可供广大自学者使用。

图书在版编目(CIP)数据

JSP开发应用项目教程/张君华主编. --北京：清华大学出版社，2015
高职高专计算机教学改革新体系规划教材
ISBN 978-7-302-40243-5

Ⅰ. ①J… Ⅱ. ①张… Ⅲ. ①JAVA语言－程序设计－高等职业教育－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2015)第109622号

责任编辑：刘士平
封面设计：傅瑞学
责任校对：李　梅
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　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
课 件 下 载：http://www.tup.com.cn，010-62795764

印 装 者：三河市少明印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：10　　**字　　数**：224千字
版　　次：2015年6月第1版　　**印　　次**：2015年6月第1次印刷
印　　数：1～2000
定　　价：25.00元

产品编号：064820-01

前言

FOREWORD

JSP 技术作为基于 Java 的动态网页设计技术，得到业内的广泛认可。本书是作者在 JSP 课程的教学实践中，为保证最终的学习效果而进行的一个有益尝试。全书围绕开发一个社区居家养老服务信息系统展开讨论，在系统开发过程中逐一介绍 JSP 技术的核心内容。

本书主要内容包括：系统开发准备、系统首页设计、管理员首页与系统管理设计、老年客户管理、社工信息处理、工单信息处理、信息查询与统计、信息发布与维护等。各章由新知识预备、系统开发、新知识使用案例剖析、系统进一步开发与实训等部分构成。在“系统开发”部分给出完整代码，以便学习者学习、模仿。在“系统进一步开发”部分仅给出部分代码，或者系统设计思路，让学习者在独立思考的基础上完成程序设计。

与现有各类 JSP 技术教材相比，本书最大的特点是从一开始就在一个真实的项目环境中开展教学，不仅给学习者提供了可以模仿、借鉴的部分，还给学习者独立思考和练习的机会；学习者在学完本书后，不仅可以掌握 JSP 技术的核心内容，而且获得了大量的代码编写训练，实现从生手到熟手的转变，还能体验一个完整项目的开发过程，避免将大量的时间用于零星的知识点学习。

本书可作为高职高专院校、各类培训机构的 JSP 学习教材，也可供广大自学者使用。

本书由张君华主编。感谢郑哲、郭双宙等老师在本书编写过程中提供的帮助，也感谢宁波城市职业技术学院的大力支持。

由于编者水平有限，不当之处恳请读者批评指正。编者的电子邮箱是 zhangjunhua_cn@163.com，欢迎来信交流。

编　者

2015 年 5 月

目录

CONTENTS

第 1 章　系统开发准备 …… 1

1.1　知识预备 …… 1

JSP 文件的构成 …… 1

1.2　开发环境准备 …… 3

1.3　知识解析 …… 6

1.3.1　JSP 的工作流程 …… 6

1.3.2　JSP 中的声明 …… 9

1.4　系统功能总体介绍 …… 10

1.5　实训安排 …… 11

第 2 章　系统首页设计 …… 12

2.1　知识预备 …… 12

2.1.1　Servlet …… 12

2.1.2　request …… 12

2.2　系统首页设计 …… 13

2.2.1　信息处理流程 …… 13

2.2.2　数据库设计 …… 13

2.2.3　系统架构设计 …… 14

2.2.4　界面设计 …… 14

2.2.5　创建控制器 Servlet …… 16

2.2.6　创建数据 Bean …… 18

2.2.7　创建处理 Bean …… 19

2.2.8　验证登录信息 …… 23

2.3　知识解析 …… 25

2.3.1　使用 Servlet …… 25

2.3.2　使用 request …… 26

2.4　实训安排 …… 26

第 3 章　管理员首页与系统管理设计 …… 28

3.1　知识预备 …… 28

3.1.1 session …… 28
3.1.2 out …… 28
3.2 管理员首页设计 …… 29
3.2.1 管理员首页的框架设计 …… 29
3.2.2 管理员首页的代码编写 …… 30
3.3 系统管理设计 …… 32
3.3.1 退出系统与重新登录 …… 32
3.3.2 系统安全性设计 …… 33
3.4 知识解析 …… 34
3.4.1 使用 session …… 34
3.4.2 使用 out …… 35
3.5 实训安排 …… 35
第 4 章 老年客户管理 …… 36
4.1 知识预备 …… 36
4.1.1 JavaBean …… 36
4.1.2 数据库操作 …… 36
4.2 客户信息登记 …… 37
4.2.1 信息处理流程 …… 37
4.2.2 数据库设计 …… 37
4.2.3 界面设计 …… 38
4.2.4 控制器设计 …… 39
4.3 知识解析 …… 45
4.3.1 JavaBean 的使用 …… 45
4.3.2 数据库的操作 …… 46
4.4 客户信息变更 …… 46
4.4.1 信息处理流程 …… 46
4.4.2 界面设计 …… 47
4.4.3 控制器设计 …… 49
4.5 实训安排 …… 53
第 5 章 社工信息处理 …… 54
5.1 知识预备 …… 54
5.1.1 JSP 表达式 …… 54
5.1.2 response …… 54
5.2 社工信息登记 …… 54
5.2.1 信息处理流程 …… 54
5.2.2 数据库设计 …… 54

5.2.3 界面设计 …… 55
5.2.4 设计控制器 SocialerAdd …… 61
5.3 知识解析 …… 64
5.3.1 使用 JSP 表达式 …… 64
5.3.2 使用 response …… 64
5.4 社工信息变更 …… 64
5.4.1 信息处理流程 …… 64
5.4.2 界面设计 …… 65
5.4.3 控制器设计 …… 66
5.5 实训安排 …… 66
第 6 章 工单信息处理 …… 67
6.1 知识预备 …… 67
6.1.1 Page 指令 …… 67
6.1.2 程序调试 …… 67
6.2 工单生成 …… 68
6.2.1 信息处理流程 …… 68
6.2.2 数据库设计 …… 68
6.2.3 界面设计 …… 69
6.2.4 控制器设计 …… 70
6.3 知识解析 …… 76
6.3.1 使用 page 指令 …… 76
6.3.2 程序调试 1：借助服务器的提示直接发现代码错误 …… 76
6.3.3 程序调试 2：借助 MyEclipse 的 Debug 功能发现代码错误 …… 77
6.4 工单完成 …… 79
6.4.1 信息处理流程 …… 79
6.4.2 界面设计 …… 80
6.4.3 控制器设计 …… 81
6.5 工单查询 …… 81
6.5.1 信息处理流程 …… 81
6.5.2 界面设计 …… 82
6.5.3 设计控制器 ServlistSearch …… 83
6.6 实训安排 …… 87
第 7 章 信息查询与统计 …… 88
7.1 知识预备 …… 88
7.1.1 创建文件与目录 …… 88
7.1.2 使用字符流读写文件 …… 88

7.2 客户费用清单 …… 89
7.2.1 信息处理流程 …… 89
7.2.2 界面设计 …… 89
7.2.3 设计控制器 FeeList …… 95
7.3 知识解析 …… 107
创建文件与目录 …… 107
7.4 其他查询 …… 108
7.4.1 信息处理流程 …… 108
7.4.2 界面设计 …… 108
7.4.3 设计控制器 MixedList …… 110
7.5 实训安排 …… 118
第 8 章 信息发布与维护 …… 119
8.1 知识预备 …… 119
application …… 119
8.2 信息发布 …… 119
8.2.1 信息处理流程 …… 119
8.2.2 数据库设计 …… 120
8.2.3 界面设计 …… 121
8.2.4 设计控制器 MessageRelease …… 122
8.3 知识解析 …… 125
使用 application …… 125
8.4 信息维护 …… 126
8.4.1 信息处理流程 …… 126
8.4.2 界面设计 …… 126
8.4.3 控制器设计 …… 130
8.5 系统首页设计与信息显示 …… 137
8.5.1 界面设计 …… 137
8.5.2 信息显示 …… 140
8.5.3 设计控制器 MsgDisplay …… 141
8.6 实训安排 …… 143
附录 1 JSP 基础知识索引 …… 144
附录 2 系统安装 …… 145
附录 3 报表生成技术 …… 147
参考文献 …… 149

第1章 系统开发准备

Chapter 1

1.1 知识预备

JSP文件的构成

一个JSP文件由HTML标记、JSP标签和JSP脚本组成。HTML代码负责页面的显示部分，后两者负责生成页面上的动态内容，使内容的生成与显示分离。

JSP标签包括指令标签和动作标签。JSP指令标签有Page指令、Include指令等。JSP动作标签有jsp:include、jsp:forward、jsp:getProperty等。通过这些标签，可以灵活地设置JSP文件或其中的对象。JSP脚本部分包括变量和方法的声明、Java程序片和Java表达式。

下面是一个JSP文件的代码。

```
1  <%@ page language="java" contentType="text/html; charset=GB18030"pageEncoding=
   "GB18030"%>
2  <%@ page import="table.messagerelease.MessageReleaseTbl" %>
3  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/
   html4/loose.dtd">
4  <html>
5    <head>
6      <meta http-equiv="Content-Type" content="text/html; charset=GB18030">
7      <title>信息编辑</title>
8      <style type="text/css">
9        #apDiv1 {
10         position:absolute;
11         width:89px;
12         height:19px;
13         z-index:1;
14         left: 455px;
15         top: 34px;
16       }
17       #apDiv2 {
18         position:absolute;
19         width:617px;
20         height:274px;
21         z-index:2;
```

```
22          left: 302px;
23          top: 68px;
24        }
25        #apDiv3 {
26          position:absolute;
27          width:74px;
28          height:22px;
29          z-index:3;
30          left: 583px;
31          top: 380px;
32        }
33      </style>
34    </head>
35    <body>
36      <%
37        MessageReleaseTbl msg = (MessageReleaseTbl)request.getAttribute("msg");
                                                          //获取一条消息已输入的信息
38      %>
39      <div id="apDiv1">信息编辑</div>
40      <form id="form_messageEdit" name="form_messageEdit" method="post" action=
        "MsgEdit02">
41        <div id="apDiv2">
42        <p align="left">信息编号:<input name="messageNo" value=<% =msg.getMessageNo()
          %> readonly type="text" size="40" /></p>
43        <p align="left">信息标题:<input name="messageTitle" value=<% =msg
          .getMessageTitle() %> type="text" size="40" /></p>
44        <p align="left">信息类别:<select name="messageSort">
45          <option value=1 <% if(msg.getMessageSort()==1){ %>selected<% } %>>常规
            信息</option>
46          <option value=2 <% if(msg.getMessageSort()==2){ %>selected<% } %>>再就
            业信息</option>
47          <option value=3 <% if(msg.getMessageSort()==3){ %>selected<% } %>>常用
            电话与联系人</option>
48          <option value=4 <% if(msg.getMessageSort()==4){ %>selected<% } %>>养生
            保健网站</option>
49          <option value=5 <% if(msg.getMessageSort()==5){ %>selected<% } %>>网站
            版权</option>
50          /select>
51        </p>
52        <p><textarea name="messageContent" cols="85" rows="10"><% =msg.getMessageContent
          ()%></textarea></p>
53        </div>
54        <div id="apDiv3"><input type="submit" name="button" id="button" value="修
          改完成" /></div>
55      </form>
56    </body>
57  </html>
```

该文件的主体部分是 html 文件的内容，包含 html、title、head、body 等标记。但在

第36～38行，通过“<%”与“%>”嵌入了Java程序片；在第42～49行等，通过“<%=”与“%>”嵌入了JSP表达式。这些属于JSP脚本。在第1、第2行，通过“<%@ page”与“%>”嵌入了JSP指令标签。在第37行，通过“//”，加入了程序的注释性说明。

1.2 开发环境准备

实现一个JSP应用开发项目，需要安装支持Java程序运行的软件包JDK或JRE，安装解释和执行JSP程序的服务器如Tomcat，以及JSP集成开发环境MyEclipse。项目开发一般需要用到数据库，因此要安装数据库系统。本节以网站系统开发常用的MySQL数据库为例来介绍。为支持对MySQL数据库的图形化管理，还要安装其管理工具Navicate Lite。为支持网页的高效设计，还要安装Dreamweaver软件。

1. 安装JRE

下载并安装软件包jdk-7u40-windows-i586，然后根据向导提示安装。记下安装位置：C:\Program Files (x86)\Java\jre7\。

2. 安装Tomcat

下载压缩包apache-tomcat-6.0.18.zip并解压缩，然后将解压后的文件夹apache-tomcat-6.0.18及其下属子目录和文件一起拷贝到C盘。

为了简化，可将C:\apache-tomcat-6.0.18\ apache-tomcat-6.0.18\...的形式调整为C:\apache-tomcat-6.0.18\...。

3. 安装MyEclipse

下载软件包myeclipse.8.5.0-win32，然后根据向导提示安装。可下载软件包myeclipse8.5_KeyGen，利用其产生的用户名和序列号进行注册。

运行MyEclipse，在其菜单MyEclipse项选中Preferences，在弹出的Preferences界面中选择MyEclipse\Servers\Tomcat\Tomcat 6.x。对Tomcat 6.x下的JDK进行设置，如图1-1所示。其中，JRE的路径即其安装路径。单击Finish按钮完成设置。

接下来，设置Tomcat服务器本身。选中Tomcat 6.x，对右侧界面进行设置，如图1-2所示。其中，Tomcat home directory设置为存放Tomcat的主目录。

4. 测试MyEclipse基本开发环境

在MyEclipse中，选择File\New\Web Project\，创建一个New Web Project。然后，输入Project Name：Test，再单击Finish按钮。

在快捷图标中，选择Deploy MyEclipse J2EE Project to Server...，将Test项目发布到服务器。其中，服务器选择Tomcat 6.x。增加的新工程发布设置如图1-3所示。

在快捷图标中，单击Run\Stop\Restart MyEclipse Servers右侧的倒三角图标，然后选择Tomcat 6.x\Start\，开启Tomcat服务器。

图 1-1 JRE 设置

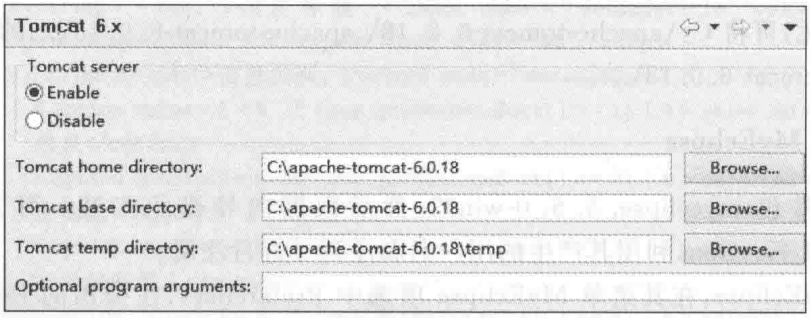

图 1-2 Tomcat 设置

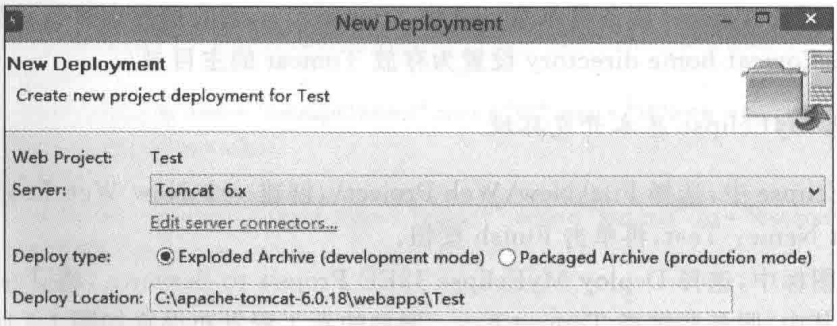

图 1-3 工程发布设置

打开 IE 浏览器，在其地址栏输入 http://localhost:8080/Test/，并按回车键。网页内容为 This is my JSP page。关闭 Tomcat 服务器，完成测试。至此，基本开发环境搭建成功。

5. 安装 MySQL 数据库系统

下载软件包 mysql-5.1.62-win32，双击运行安装向导。选择接受许可证协议、典型安装(Setup Type：Typical)完成安装，并进一步配置。在对 MySQL Server Instance 的配置中，依次选择 Detailed Configuration→Developer Machine→Multifunctional Database→Decision Support (DSS)/OLAP→Enable TCP/IP Networking (Port Number：3306)/Enable Strict Mode→Manual Selected Default Character Set/Collation(Character Set：gbk)→Install As Windows Service(Service Name：MySQL→Lauch the MySQL Server automatically)。

在 Security Settings 部分，输入密码 1234 并确认；选择 Enable root access from remote machines，允许远程访问本数据库系统，如图 1-4 所示。记下该密码，在数据库部分的程序设计中要用到它。

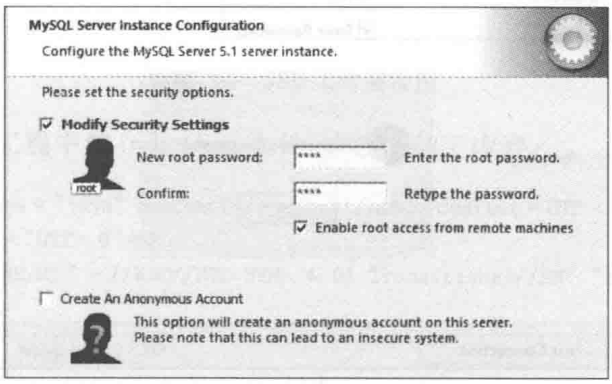

图 1-4 数据库系统安全设置

最后选择 Execute，弹出界面如图 1-5 所示，表示数据库系统安装成功。单击 Finish 按钮，完成 MySQL 的安装。

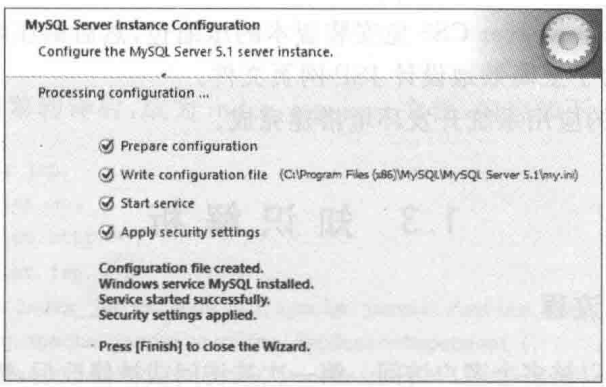

图 1-5 数据库系统安装完成

6. 安装 MySQL 管理工具 Navicate Lite

下载软件包 navicat100_lite_en_87700.zip，双击安装程序，根据向导提示完成软件安装。

7. Navicate Lite 与 MySQL 数据库的连通性测试

运行 Navicate Lite，选择 File\New Connection\MySQL\。在 Connection 的配置界面中输入连接名 MySqlConn，再输入 root 对应的密码 1234，然后单击左下角的按钮 Test Connection，显示 Connection Successful，表示 Navicate Lite 与 MySQL 数据库连接成功，如图 1-6 所示。

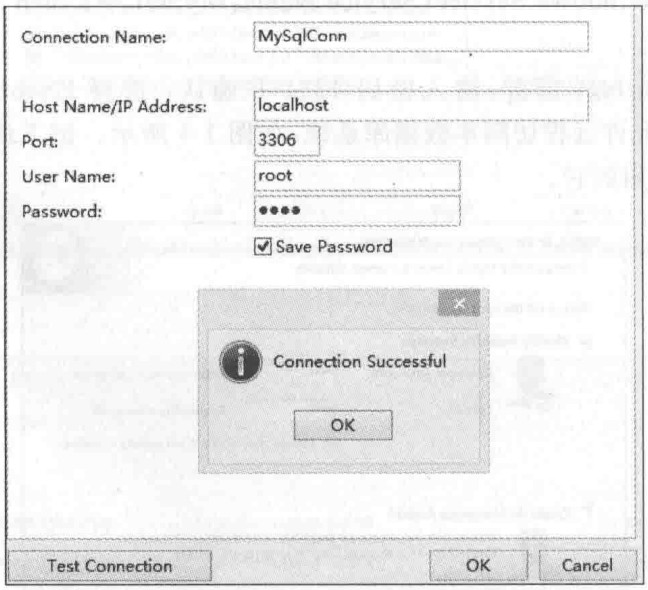

图 1-6 Navicate Lite 与 MySQL 的连通性测试

还剩下 MyEclipse 与 MySQL 的连通性测试，留待后续章节完成。

8. 安装 Dreamweaver CS5

下载 Adobe Dreamweaver CS5 免安装版本的压缩包，然后解压缩并存放于 C 盘。安装 Deamweaver，是为了更高效地设计 JSP 网页文件。

至此，基于 JSP 的应用系统开发环境搭建完成。

1.3 知识解析

1.3.1 JSP 的工作流程

一个 JSP 页面可以被多个客户访问。第一次被访问或被修改后，需要经过转译和编译，才能被执行，其他情况下可直接执行编译后的字节码。JSP 工作流程图如图 1-7 所示。

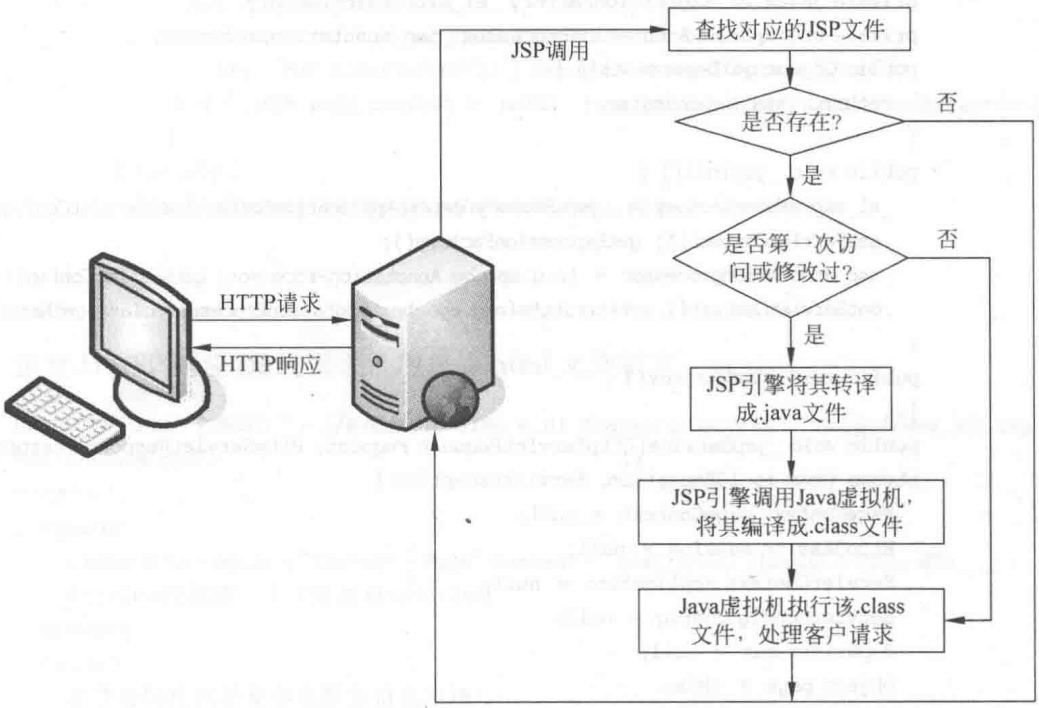

图 1-7 JSP 工作流程图

改写上述 Test 工程中的 index.jsp 文件，修改为如下内容：

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/
html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>我的第一个 JSP 文件</title>
  </head>
  <body>
    <% out.print("欢迎访问社区居家养老服务信息系统!"); %>
  </body>
</html>
```

该文件经 JSP 引擎转译后，成为 index_jsp.java 文件，内容如下。

```
package org.apache.jsp;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
public final class index_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {
        private static final JspFactory _jspxFactory = JspFactory.getDefaultFactory();
        private static java.util.List _jspx_dependants;
```

```
private javax.el.ExpressionFactory _el_expressionfactory;
private org.apache.AnnotationProcessor _jsp_annotationprocessor;
public Object getDependants() {
  return _jspx_dependants;
}
public void _jspInit() {
  _el_expressionfactory = _jspxFactory.getJspApplicationContext(getServletConfig()
  .getServletContext()).getExpressionFactory();
  _jsp_annotationprocessor = (org.apache.AnnotationProcessor) getServletConfig()
  .getServletContext().getAttribute(org.apache.AnnotationProcessor.class.getName());
}
public void _jspDestroy() {
}
public void _jspService(HttpServletRequest request, HttpServletResponse response)
throws java.io.IOException, ServletException {
  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 {
    response.setContentType("text/html; charset=UTF-8");
    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("<!DOCTYPE html PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\"
    \"http://www.w3.org/TR/html4/loose.dtd\">\r\n");
    out.write("<html>\r\n");
    out.write("<head>\r\n");
    out.write("<meta http-equiv=\"Content-Type\" content=\"text/html;
    charset=UTF-8\">\r\n");
    out.write("<title>我的第一个 JSP 文件</title>\r\n");
    out.write("</head>\r\n");
    out.write("<body>\r\n");
    out.print("欢迎访问社区居家养老服务信息系统!");
    out.write("\r\n");
    out.write("</body>\r\n");
    out.write("</html>");
  } catch (Throwable t) {
    if (!(t instanceof SkipPageException)){
```

```
                out = _jspx_out;
                if (out != null && out.getBufferSize() != 0)
                  try { out.clearBuffer(); } catch (java.io.IOException e) {}
                if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
              }
            } finally {
                _jspxFactory.releasePageContext(_jspx_page_context);
            }
        }
    }
```

作为 HTTP 响应,最终发送给用户的 html 文件如下。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/
html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>我的第一个 JSP 文件</title>
  </head>
  <body>
    欢迎访问社区居家养老服务信息系统!
  </body>
</html>
```

该 html 文件经 IE 浏览器解释后,用户看到的内容如图 1-8 所示。

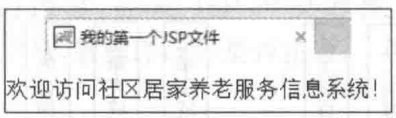

图 1-8 JSP 运行结果示例

1.3.2 JSP 中的声明

在 JSP 程序中可以预先申明变量、方法和类,然后在执行信息处理时使用。其定义的语法是:

```
<%! Declaration; [declaration;] ... %>
```

将 index.jsp 的<body>部分替换为如下代码:

```
<! public String getDayWeek(int n)
{
    String week[] = {"星期一", "星期二", "星期三", "星期四", "星期五", "星期六", "星期日"};
    return week[n];
}
 %>
<%
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(new Date());
```