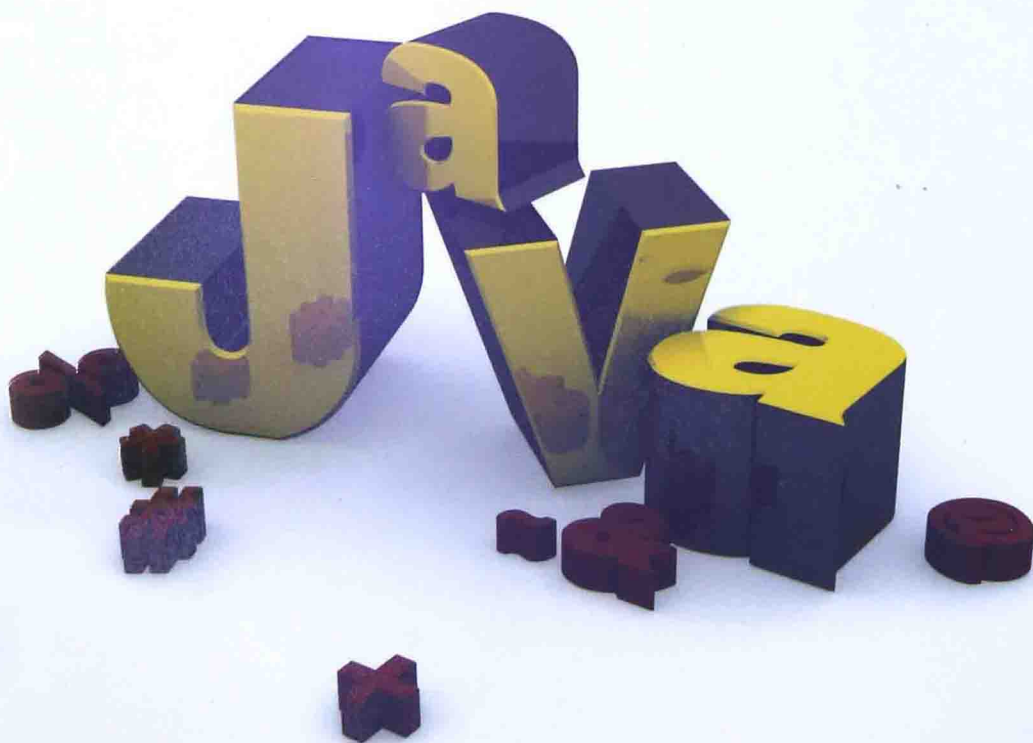


Java 高等学校Java课程系列教材

JSP 程序设计 (第2版)

上机实验与综合实训

耿祥义 张跃平 编著



清华大学出版社

高等学校 Java 课程系列教材

JSP 程序设计(第 2 版) 上机实验与综合实训

耿祥义 张跃平 编著

清华大学出版社
北 京

内 容 简 介

本书是与《JSP 程序设计(第2版)》(清华大学出版社)配套的上机实验与综合实训辅导教材。本书第一部分包括8次上机实践内容,通过一系列实验练习使学生巩固所学的知识。每个实验主要由相关知识、实验目的、实验要求、效果示例和参考代码组成。在进行实验之前,首先通过实验目的了解实验要完成的关键主题,通过实验要求知道本实验应达到怎样的标准。本书第二部分包括两个完整的Web程序设计项目,属于综合实训,其目的是掌握一般Web应用中常用基本模块的开发方法。本书的附录为《JSP 程序设计(第2版)》的习题参考解答。

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 程序设计(第2版)上机实验与综合实训/耿祥义,张跃平编著.--北京:清华大学出版社,2014

高等学校 Java 课程系列教材

ISBN 978-7-302-37237-0

I. ①J… II. ①耿… ②张… III. ①Java 语言—网页制作工具—教学参考资料 IV. ①TP312
②TP393.092

中国版本图书馆 CIP 数据核字(2014)第 153411 号

责任编辑:魏江江 王冰飞

封面设计:杨 兮

责任校对:焦丽丽

责任印制:宋 林

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>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:9.75

字 数:235 千字

版 次:2014 年 9 月第 1 版

印 次:2014 年 9 月第 1 次印刷

印 数:1~2000

定 价:24.00 元

前 言

本书是与《JSP 程序设计(第 2 版)》(清华大学出版社)配套的上机实验与综合实训辅导教材,适合初学 JSP 的读者参考使用。

本书第一部分包括 8 次上机实践内容,目的是通过一系列实验练习使学生巩固所学的知识。每个实验由五部分构成:

- (1) 相关知识点。这一部分给出与该实验相关的重点知识和难点知识。
- (2) 实验目的。让学生了解本实验需要掌握哪些知识,实验内容将以这些知识为中心。
- (3) 实验要求。给出了该实验需要达到的基本标准。
- (4) 效果示例。通过效果图,使得学生首先对本次实验有一个直观的认识。
- (5) 参考代码。给出的参考代码有一定的启发作用,学生可根据参考代码编写代码。

本书第二部分包括两个完整的 Web 程序设计项目,属于综合实践训练,其目的是让学生掌握一般 Web 应用中常用基本模块的开发方法。

可以登录清华大学出版社网站(www.tup.com.cn)下载实验模板的源程序。

编者

2014 年 4 月

目 录

第 1 章 JSP 简介(实验)	1
实验 Tomcat 服务器的安装与配置	1
第 2 章 JSP 页面与 JSP 标记(实验)	3
实验 1 JSP 页面的基本结构	3
实验 2 JSP 指令标记	5
实验 3 JSP 动作标记	6
第 3 章 Tag 文件与 Tag 标记(实验)	9
实验 1 使用标记体	9
实验 2 使用 attribute 指令	10
实验 3 使用 variable 指令	13
第 4 章 JSP 内置对象(实验)	15
实验 1 request 对象	15
实验 2 response 对象	17
实验 3 session 对象	19
第 5 章 JSP 中的文件操作(实验)	22
实验 1 使用文件字节流读/写文件	22
实验 2 使用文件字符流加密文件	26
实验 3 使用数据流读/写 Java 数据	30
第 6 章 在 JSP 中使用数据库(实验)	33
实验 1 连接查询 MySQL 数据库	33
实验 2 连接查询 SQL Server 数据库	36
实验 3 连接查询 Access 数据库	38
第 7 章 JSP 与 JavaBean(实验)	42
实验 1 有效范围为 request 的 bean	42
实验 2 有效范围为 session 的 bean	44

实验 3 有效范围为 application 的 bean	46
-------------------------------------	----

第 8 章 JavaServlet 与 MVC 模式(实验)	50
--------------------------------------	----

实验 计算两个正数的代数平均值与几何平均值	50
-----------------------------	----

第 9 章 “星星”书屋(综合实训)	55
--------------------------	----

9.1 系统主要模块	55
------------------	----

9.2 数据库设计与连接	56
--------------------	----

9.2.1 数据库设计	56
-------------------	----

9.2.2 连接数据库	57
-------------------	----

9.3 系统管理	58
----------------	----

9.4 用户注册	59
----------------	----

9.5 会员登录	62
----------------	----

9.6 浏览图书信息	64
------------------	----

9.7 查询图书	68
----------------	----

9.8 查看购物车	71
-----------------	----

9.9 订单预览	74
----------------	----

9.10 确认订单	76
-----------------	----

9.11 查看订单	79
-----------------	----

9.12 查看图书摘要	82
-------------------	----

9.13 修改密码	84
-----------------	----

9.14 修改注册信息	86
-------------------	----

9.15 退出登录	89
-----------------	----

第 10 章 网络交友(综合实训)	91
-------------------------	----

10.1 系统模块构成	91
-------------------	----

10.2 数据库设计与连接	91
---------------------	----

10.3 系统管理	93
-----------------	----

10.3.1 页面管理	93
-------------------	----

10.3.2 JavaBean 与 Servlet 管理	95
------------------------------------	----

10.3.3 配置文件	95
-------------------	----

10.4 会员注册	97
-----------------	----

10.4.1 模型	97
-----------------	----

10.4.2 控制器	98
------------------	----

10.4.3 视图(JSP 页面)	100
-------------------------	-----

10.5 会员登录	102
-----------------	-----

10.5.1 模型	102
-----------------	-----

10.5.2 控制器	103
------------------	-----

10.5.3 视图	105
-----------------	-----

10.6 上传照片	106
-----------------	-----

10.6.1 模型 106

10.6.2 控制器 107

10.6.3 视图 109

10.7 浏览会员信息 111

10.7.1 模型 111

10.7.2 控制器 113

10.7.3 视图 116

10.8 修改密码 118

10.8.1 模型 118

10.8.2 控制器 119

10.8.3 视图 121

10.9 修改注册信息 122

10.9.1 模型 122

10.9.2 控制器 123

10.10 退出登录..... 126

附录 A 习题解答..... 128

第 1 章

JSP 简介(实验)

第 1 章只有一个实验,目的是让学生熟悉 Tomcat 服务器安装与配置,为后续的实验做好准备工作。

实验 Tomcat 服务器的安装与配置

1. 相关知识点

- (1) 安装的 Tomcat 版本为 tomcat-6.0.13。
- (2) 执行 Tomcat 安装根目录中 bin 文件夹中的 startup.bat 或 tomcat6.exe 来启动 Tomcat 服务器。
- (3) JSP 页面文件保存到 Tomcat 服务器的某个 Web 服务目录中,以便远程的客户使用浏览器访问该 Tomcat 服务器上的 JSP 页面。

2. 实验目的

本实验的目的是让学生掌握怎样设置 Web 服务目录,怎样访问 Web 服务目录下的 JSP 页面,怎样修改 Tomcat 服务器的端口号。

3. 实验要求

- (1) 将下载的 apache-tomcat-6.0.13.zip 解压到硬盘某个分区,例如 D。
- (2) 在硬盘分区 D 下新建一个目录,名字为 student,将 student 设置为 Web 服务目录,并为该 Web 服务目录指定名字为 good 的虚拟目录。打开 Tomcat 安装目录中 conf 文件夹里的 server.xml 文件,找到出现</Host>的部分(server.xml 文件尾部)。然后在</Host>的前面加入:

```
<Context path = "/good" docBase = "d:/student" debug = "0" reloadable = "true" />
```

- (3) 修改端口号为 5678。在 server.xml 文件中找到修改端口号部分,将端口号修改为 5678。
- (4) 启动 Tomcat 服务器(如果已经启动,必须关闭 Tomcat 服务器,并重新启动)。如果没有按照步骤(2)的要求在 D 盘下建立名字为 student 的目录, Tomcat 将无法启动。
- (5) 用文本编辑器编写一个简单的 JSP 页面 biao.jsp,并保存到 Web 服务目录 student 中。
- (6) 用浏览器访问 Web 服务目录 student 中的 JSP 页面 biao.jsp。

4. JSP 页面效果示例

根据实验要求,必须在浏览器的地址栏中输入 Tomcat 服务器的 IP 地址和端口号,并

通过虚拟目录 good 访问 Web 服务目录 student 下的 JSP 页面。如果浏览器和 Tomcat 服务器驻留在同一计算机上,IP 地址可以是 127.0.0.1。biao.jsp 运行效果如图 1-1 所示。

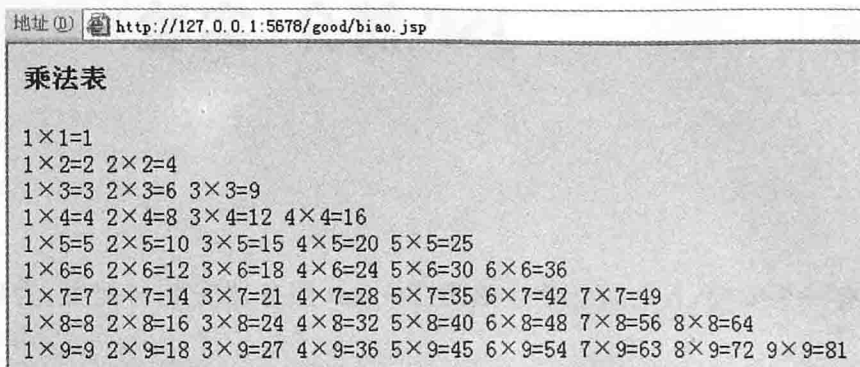


图 1-1 简单的 JSP 页面

5. 参考代码

可以按照实验要求,参考本代码编写自己的实验代码。

JSP 页面参考代码如下。

biao.jsp(效果如图 1-1 所示)

```
<% @ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY BGCOLOR = yellow>
<h3> 乘法表</h3>
<FONT Size = 3>
  <%
    for(int j = 1;j <= 9;j++){
      for(int i = 1;i <= j;i++) {
        int n = i * j;
        out.print(i + " × " + j + " = " + n + " ");
      }
      out.print("<br>");
    }
  %>
</FONT></BODY></HTML>
```

第 2 章

JSP 页面与 JSP 标记(实验)

第 2 章共有 3 个实验。要求将 Tomcat 服务器的端口号恢复默认设置,即端口号为 8080。

在 webapps 目录下新建一个 Web 服务目录 chapter2。除特别要求外,本章实验中涉及的 JSP 页面均保存在 chapter2 中。

实验 1 JSP 页面的基本结构

1. 相关知识点

一个 JSP 页面可由普通的 HTML 标记、JSP 标记、成员变量和方法的声明、Java 程序片和 Java 表达式组成。JSP 引擎把 JSP 页面中的 HTML 标记交给客户的浏览器执行显示;JSP 引擎负责处理 JSP 标记、变量和方法声明;JSP 引擎负责运行 Java 程序片、计算 Java 表达式,并将需要显示的结果发送给客户的浏览器。

JSP 页面中的成员变量是被所有用户共享的变量。Java 程序片可以操作成员变量,任何一个用户对 JSP 页面成员变量操作的结果,都会影响到其他用户。如果多个用户访问一个 JSP 页面,那么该页面中的 Java 程序片就会被执行多次,分别运行在不同的线程中,即运行在不同的时间片内。运行在不同线程中的 Java 程序片的局部变量互不干扰,即一个用户改变 Java 程序片中的局部变量的值不会影响其他用户的 Java 程序片中的局部变量。

2. 实验目的

本实验的目的是让学生掌握怎样在 JSP 页面中使用成员变量,怎样使用 Java 程序片、Java 表达式。

3. 实验要求

本实验将用户输入的单词按字典顺序排序。需要编写两个 JSP 页面,名字分别为 inputWord.jsp 和 showDictionary.jsp。

1) inputWord.jsp 的具体要求

该页面有一个表单,用户通过该表单输入若干个单词,并提交给 showDictionary.jsp 页面。

2) showDictionary.jsp 的具体要求

该页面负责排序单词,并将排序的全部单词显示给用户。

(1) 该 JSP 页面有名字为 dictionary,类型是 TreeSet 成员变量。

(2) 该 JSP 页面有 public void addWord(String s)方法,该方法将参数 s 指定的字符串

添加到成员变量 dictionary 中。

(3) 该 JSP 页面在程序片中操作 dictionary, 即显示全部的单词。

4. JSP 页面效果示例

inputWord.jsp 的效果如图 2-1 所示。

showDictionary.jsp 的效果如图 2-2 所示。

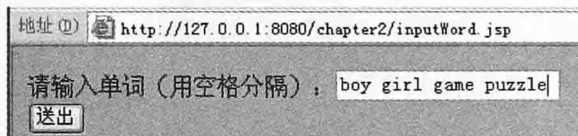


图 2-1 输入单词

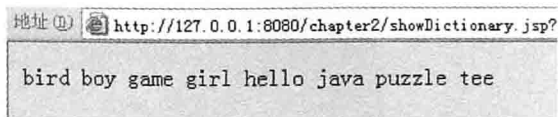


图 2-2 排序并显示单词

5. 参考代码

可以按照实验要求, 参考本代码编写自己的实验代码。

JSP 页面参考代码如下。

inputWord.jsp

```
<% @ page contentType = "text/html; charset = GB2312" %>
<HTML>
<BODY bgcolor = cyan>
<FONT size = 3>
    <FORM action = "showDictionary.jsp" method = get name = form>
        请输入单词(用空格分隔): <INPUT type = "text" name = "word">
        <BR><INPUT TYPE = "submit" value = "送出" name = submit>
    </FORM>
</BODY>
</HTML>
```

showDictionary.jsp

```
<% @ page contentType = "text/html; charset = GB2312" %>
<% @ page import = "java.util. *" %>
<HTML>
<BODY BGCOLOR = yellow>
<FONT Size = 3>
<% !
    TreeSet<String> dictionary = new TreeSet<String>();
    public void addWord(String s) {
        String word[] = s.split(" ");
        for(int i = 0; i < word.length; i++) {
            dictionary.add(word[i]);
        }
    }
%>
<%
    String str = request.getParameter("word");
    addWord(str);
    Iterator<String> te = dictionary.iterator();
    while(te.hasNext()) {
```

```

String word = te.next();
out.print(" " + word);
}
%>
</FONT>
</BODY>
</HTML>

```

实验 2 JSP 指令标记

1. 相关知识点

include 指令标记：`<%@ include file = "文件的 URL " %>`的作用是在 JSP 页面出现该指令的位置处，静态插入一个文件。被插入的文件必须是可访问和可使用的，如果该文件和当前 JSP 页面在同一 Web 服务目录中，那么“文件的 URL”就是文件的名称；如果该文件在 JSP 页面所在 Web 服务目录的一个子目录中，例如 fileDir 子目录中，那么“文件的 URL”就是“fileDir/文件的名称”。include 指令标记是在编译阶段就处理所需要的文件，被处理的文件在逻辑和语法上依赖于当前 JSP 页面，其优点是页面的执行速度快。

2. 实验目的

本实验的目的是让学生掌握怎样在 JSP 页面中使用 include 指令标记在 JSP 页面中静态插入一个文件的内容。

3. 实验要求

该实验要求使用 include 指令标记使得每个页面都包含有导航条。在进行实验之前，将名字是 leader.txt 的文件保存到本实验所使用的 web 服务目录中。leader.txt 的内容如下：

leader.txt

```

<% @ page contentType = "text/html; charset = GB2312" %>
<a href = "first.jsp"> 链接到页面 1 </a>
<a href = "second.jsp"> 链接到页面 2 </a>
<a href = "third.jsp"> 链接到页面 3 </a>

```

实验要求编写 3 个 JSP 页面，具体要求如下。

1) first.jsp 的具体要求

first.jsp 使用 include 指令静态插入 leader.txt 文本文件。

2) second.jsp 的具体要求

second.jsp 使用 include 指令静态插入 leader.txt 文件。

3) third.jsp 的具体要求

third.jsp 使用 include 指令静态插入 leader.txt。

4. JSP 页面效果示例

first.jsp 的效果如图 2-3 所示。

second.jsp 的效果如图 2-4 所示。

third.jsp 的效果如图 2-5 所示。

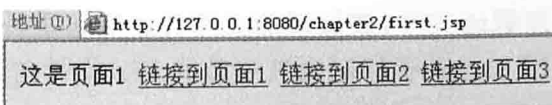


图 2-3 first.jsp 页面中的导航条

地址 ① http://127.0.0.1:8080/chapter2/second.jsp

这是页面2 [链接到页面1](#) [链接到页面2](#) [链接到页面3](#)

图 2-4 second.jsp 页面中的导航条

地址 ① http://127.0.0.1:8080/chapter2/third.jsp

这是页面3 [链接到页面1](#) [链接到页面2](#) [链接到页面3](#)

图 2-5 third.jsp 页面中的导航条

5. 参考代码

可以按照实验要求,参考本代码编写自己的实验代码。

JSP 页面参考代码如下。

first.jsp

```
<% @ page contentType = "text/html; charset = GB2312" %>
<HTML>
< BODY BGCOLOR = yellow>
< P>这是页面 1
<% @include file = "leader.txt" %>
</BODY>
</HTML>
```

second.jsp

```
<% @ page contentType = "text/html; charset = GB2312" %>
<HTML>
< BODY BGCOLOR = cyan>
< P>这是页面 2
<% @include file = "leader.txt" %>
</BODY>
</HTML>
```

third.jsp

```
<% @ page contentType = "text/html; charset = GB2312" %>
<HTML>
< BODY BGCOLOR = green>
< P>这是页面 3
<% @include file = "leader.txt" %>
</BODY>
</HTML>>
```

实验 3 JSP 动作标记

1. 相关知识点

include 动作标记: `<jsp:include page="文件的 URL"/>`是在 JSP 页面运行时才处理加载的文件,被加载的文件在逻辑和语法上独立于当前 JSP 页面。include 动作标记可以使用 param 子标记向被加载的 JSP 文件传递信息。

forward 动作标记 `<jsp:forward page="要转向的页面" />`作用是从该指令处停止当前页面的继续执行,而转向执行 page 属性指定的 JSP 页面。forward 标记可以使用 param

动作标记作为子标记,以便向要转向的 JSP 页面传送信息。

2. 实验目的

本实验的目的是让学生掌握怎样在 JSP 页面中使用 include 标记动态加载文件,使用 forward 实现页面的转向。

3. 实验要求

编写 3 个 JSP 页面: giveFileName.jsp、readFile.jsp 和 error.jsp。

1) giveFileName.jsp 的具体要求

要求 giveFileName.jsp 页面使用 include 动作标记动态加载 readFile.jsp 页面,并将一个文件的名称比如 ok.txt 传递给被加载的 readFile.jsp 页面。

2) readFile.jsp 的具体要求

要求 readFile.jsp 负责根据 giveFileName.jsp 页面传递过来的文件名字进行文件的读取操作,如果该文件不存在就使用 forward 动作标记将用户转向 error.jsp 页面。

3) error.jsp 的具体要求

负责显示错误信息。

4. JSP 页面效果示例

giveFileName.jsp 的效果如图 2-6 所示。

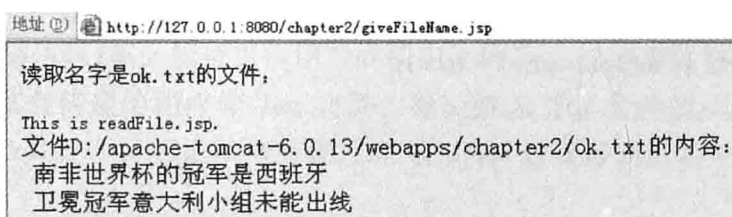


图 2-6 使用 include 动作标记加载 readFile.jsp

readFile.jsp 的效果如图 2-7 所示。

error.jsp 的效果如图 2-8 所示。

```
This is readFile.jsp.  
文件D:/apache-tomcat-6.0.13/webapps/chapter2/ok.txt的内容:  
南非世界杯的冠军是西班牙  
卫冕冠军意大利小组未能出线
```

图 2-7 根据文件名字读取文件的内容

```
This is error.jsp.  
本页面得到的信息:File Not Found
```

图 2-8 显示错误信息

5. 参考代码

可以按照实验要求,参考本代码编写自己的实验代码。

JSP 页面参考代码如下。

giveFileName.jsp

```
<% @ page contentType = "text/html; charset = GB2312" %>  
<HTML>  
<BODY bgcolor = yellow>  
    读取名字是 ok.txt 的文件:  
    <jsp:include page = "readFile.jsp">
```

```

        <jsp:param name = "file" value = "D:/apache - tomcat - 6.0.13/webapps/chapter2/ok.txt"/>
    </jsp:include>
</BODY>
</HTML>

```

readFile.jsp

```

<% @ page contentType = "text/html; charset = GB2312" %>
<% @ page import = "java.io. * " %>
<HTML>
    <BODY bgcolor = cyan>
        <P><Font size = 2 color = blue>
            This is readFile.jsp.
        </Font>
        <Font size = 4>
        <%
            String s = request.getParameter("file");
            File f = new File(s);
            if(f.exists()){
                out.println("<BR>文件" + s + "的内容: ");
                FileReader in = new FileReader(f);
                BufferedReader bIn = new BufferedReader(in);
                String line = null;
                while((line = bIn.readLine())!= null){
                    out.println("<br>" + line);
                }
            }
            else {
                %>
                <jsp:forward page = "error.jsp" >
                    <jsp:param name = "mess" value = "File Not Found" />
                </jsp:forward>
            }
        %>
    </FONT>
</BODY>
</HTML>

```

error.jsp

```

<% @ page contentType = "text/html; charset = GB2312" %>
<HTML>
    <BODY bgcolor = yellow>
        <P><Font size = 5 color = red>
            This is error.jsp.
        </Font>
        <Font size = 2>
        <%
            String s = request.getParameter("mess");
            out.println("<BR>本页面得到的信息:" + s);
        %>
    </FONT>
</BODY>
</HTML>

```

第 3 章

Tag 文件与 Tag 标记(实验)

要求在 webapps 目录下新建一个 Web 服务目录 chapter3。除特别要求外,本章实验所涉及的 JSP 页面均保存在 chapter3 中;Tag 文件保存在 chapter3\WEB-INF\tags 目录中。

实验 1 使用标记体

1. 相关知识点

Tag 文件是扩展名为 .tag 的文本文件,其结构几乎和 JSP 文件相同。一个 Tag 文件中可以有普通的 HTML 标记符、某些特殊的指令标记、成员变量和方法、Java 程序片和 Java 表达式。JSP 页面使用 Tag 标记动态执行一个 Tag 文件。当 JSP 页面调用一个 Tag 文件时希望动态地向 Tag 文件传递信息,那么就可以使用带有标记体的 Tag 标记来执行一个 Tag 文件,Tag 标记中的“标记体”就会传递给相应的 Tag 文件。标记体由 Tag 文件的<jsp:doBody/>标记负责处理,即<jsp:doBody />标记被替换成处理“标记体”后所得到的结果。

2. 实验目的

本实验的目的是让学生灵活掌握在 Tag 标记中使用标记体。

3. 实验要求

编写一个 JSP 页面 giveMess.jsp 和一个 Tag 文件 handleMess.tag。JSP 页面通过调用 Tag 文件在表格中的单元格显示文本,该 JSP 页面通过使用标记体将要显示的文本传递给被调用的 Tag 文件。

1) giveMess.jsp 的具体要求

要求 giveMess.jsp 页面使用带标记体的 Tag 标记来调用 Tag 文件,其中标记体是一行文本,如下所示:

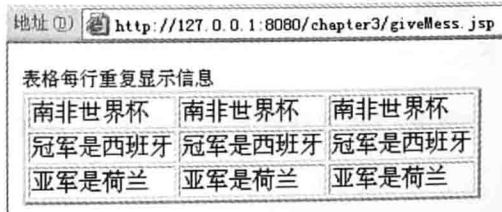
```
<ok:handleMess>  
    南非世界杯  
</ok:handleMess>
```

2) handleMess.tag 的具体要求

handleMess.tag 使用<jsp:doBody/>处理标记体,将标记体给出的文本显示在表格的单元格中。要求表格每行有 3 个单元,重复显示标记体给出的文本。

4. JSP 页面效果示例

giveMess.jsp 的效果如图 3-1 所示。



表格每行重复显示信息

南非世界杯	南非世界杯	南非世界杯
冠军是西班牙	冠军是西班牙	冠军是西班牙
亚军是荷兰	亚军是荷兰	亚军是荷兰

图 3-1 使用带标记体的 Tag 标记

5. 参考代码

可以按照实验要求,参考本代码编写自己的实验代码。

1) JSP 页面参考代码

giveMess.jsp

```
<% @ page contentType = "text/html;Charset = GB2312" %>
<% @ taglib tagdir = "/WEB-INF/tags" prefix = "ok" %>
<html>
  <body>
    <font size = 2 color = blue>表格每行重复显示信息</font>
    <table border = 2>
      <ok:handleMess>
        南非世界杯
      </ok:handleMess>
      <ok:handleMess>
        冠军是西班牙
      </ok:handleMess>
      <ok:handleMess>
        亚军是荷兰
      </ok:handleMess>
    </table>
  </body>
</html>
```

2) Tag 文件参考代码

handleMess.Tag

```
<tr>
  <td><jsp:doBody/></td>
  <td><jsp:doBody/></td>
  <td><jsp:doBody/></td>
</tr>
```

实验 2 使用 attribute 指令

1. 相关知识

一个 Tag 文件中通过使用 attribute 指令:

```
<% @ attribute name = "对象名字" required = "true"|"false" type = "对象的类型" %>
```

使得 JSP 页面在调用 Tag 文件时,可以向该 Tag 文件中的对象传递一个引用,方式如下:

```
<前缀: Tag 文件名字 对象名字 = "对象的引用" />
```

或

```
<前缀: Tag 文件名字 对象名字 = "对象的引用">
  标记体
```