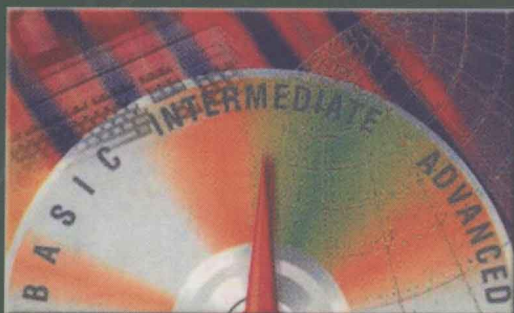


**COMP-U-LEARN**

*Setting Standards in Technology*

印度优质 IT 职业教育教学用书



# JSP 开发技术

## (指导手册)

Comp-U-Learn Tech India Ltd.

王 强 贾素玲 许 珂 张 成 编译



高 等 教 育 出 版 社  
HIGHER EDUCATION PRESS

印度优质 IT 职业教育教学用书

# JSP 开发技术

(指导手册)

Comp-U-Learn Tech India Ltd.

王 强 贾素玲 许 珂 张 成 编译

高等教育出版社

# 目 录

## 第一部分 案例题目 .....1

第一章 JSP 简介 .....1

第二章 配置 JSP 运行环境 .....1

第三章 JSP 基本语法 .....2

第四章 使用 JSP 内置对象 .....2

第五章 JSP 与 Servlet .....3

第六章 使用 JavaBean .....3

第七章 JSP 访问数据库 .....3

第八章 JSP 与 XML .....3

## 第二部分 参考答案 .....5

第一章 JSP 简介 .....5

第二章 配置 JSP 运行环境 .....6

第三章 JSP 基本语法 .....9

第四章 使用 JSP 内置对象 .....13

第五章 JSP 与 Servlet .....21

第六章 使用 JavaBean .....29

第七章 JSP 访问数据库 .....39

第八章 JSP 与 XML .....52

# 第一部分 案例题目

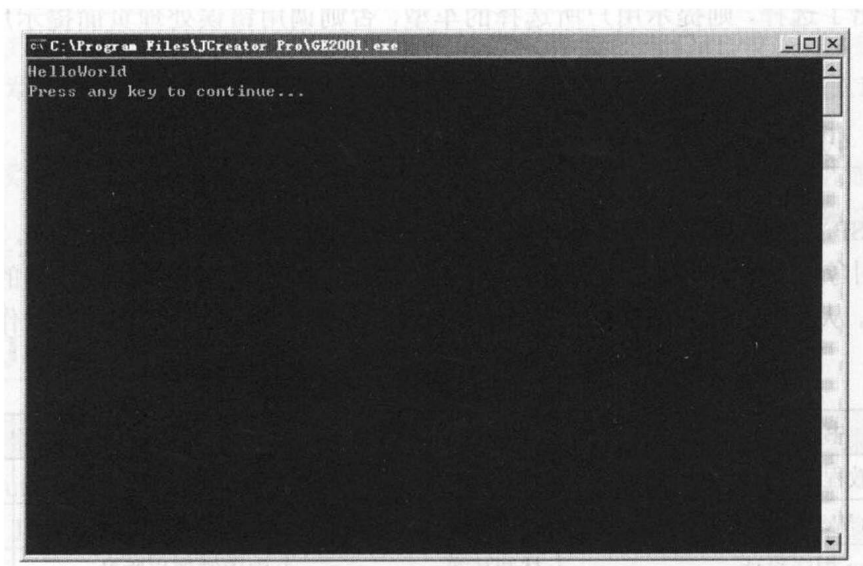
## 第一章 JSP 简介

1. 登录 Chinaren 校友录, 网址: <http://alumni.chinaren.com>。这是一个以 JSP 技术为主的网站, 你能够证明这一事实吗? 该网站还采用了 PHP、CGI 等技术, 但是没有采用 ASP 技术, 你怎么才能证明这一观点?

2. 怎样查看 <http://alumni.chinaren.com/index.jsp> 的源文件? 你所看到的源文件是 index.jsp 页面的真正源文件吗? 为什么?

## 第二章 配置 JSP 运行环境

1. 编写一个最简单的 Java 程序, 能够输出 “HelloWorld”。在 JCreator (或者其他 Java 开发工具) 环境下编译执行该文件, 以便测试你的开发工具和 JDK 是否工作正常。如果一切正常, 结果应当如题图 2.1 所示。



题图 2.1

2. 编写一个最简单的 JSP 程序，同样能够在浏览器中输出 “HelloWorld”。使用 Tomcat 5 作为 Web 服务器，要求在浏览器地址栏中输入：

http://localhost:8000/Charppter2/HelloWorld.jsp  
或 http://localhost:8080/Charppter2/HelloWorld.jsp  
即可以访问该 JSP 页面。

3. JSP 程序不变，要求在浏览器中输入 http://localhost/Charppter2/HelloWorld.jsp 即可以实现对该页面的访问。

4. JSP 程序不变，要求在浏览器中输入 http://localhost/HelloWorld.jsp 即可以实现对该页面的访问。

5. JSP 程序不变，要求输入 http://localhost/ 即可以访问该页面。

6. JSP 程序不变，要求输入 http://www.helloworld.com/ 即可以访问该页面。

### 第三章 JSP 基本语法

1. 用 JSP 实现猜数字游戏。游戏规则：给出一个 1~100 之间的随机整数，允许用户不断猜测该数，如果小于该数则提示“猜小了，请输入更大的数”，若大于该数则提示“猜大了，请输入更小的数”，若猜中则提示“恭喜！你猜中了。”

2. 编写一个 JSP 页面 currenttime.jsp，用来获取系统当前时间（以毫秒为单位，可直接通过<%= System.currentTimeMillis() %>获得），但是需要在另一个页面 main.jsp 中显示该时间。

3. 在 JSP 页面中（或 HTML 页面中）提供几种车型（如宝马、奥迪等）供用户选择，如果用户做出了选择，则提示用户所选择的车型，否则调用错误处理页面提示用户必须选择一种车型。

### 第四章 使用 JSP 内置对象

在使用 JSP 内置对象一章中我们曾经提到使用 session 可以实现购物车功能，本题的要求是利用 session 对象以及其他 JSP 内置对象实现简单的网上图书购书。至少包括图书的网上购买（添加至购物车）、从购物车删除、结账等功能。读者可以采用题表 4.1 中的图书数据作为购买依据。

题表 4.1

| 书 名                      | 作 者          | 出 版 社    | 售 价  |
|--------------------------|--------------|----------|------|
| C++程序设计教程                | 钱能           | 清华大学出版社  | 38.0 |
| Tomcat 与 Java Web 开发技术详解 | 孙卫琴          | 电子工业出版社  | 45.0 |
| 精通 Java Swing 程序设计       | 林智扬等         | 中国铁道出版社  | 50.0 |
| Java Servlet 编程          | Jason Hunter | O'REILLY | 85.0 |

## 第五章 JSP 与 Servlet

在第四章的案例练习中，笔者用 `BookCart.jsp` 实现了添加至购物车、从购物车删除等功能（参见第四章的习题答案），现在需要将这些功能封装在 `Servlet` 中实现，而 `BookCart.jsp` 等 JSP 页面专门负责图书信息或购物车信息的显示。请参照第四章原题，使用 `Servlet` 和 JSP 技术共同实现网上购书的功能。

## 第六章 使用 JavaBean

我们仍然沿用前述章节购物车的例子。在前面几章中，笔者用数组对图书简单进行了描述（参见第五章的习题答案），学习了 `JavaBean` 之后，我们就可以把图书封装成 `Bean` 对象，这样程序的独立性和可扩展性更强。请读者考虑怎样结合使用 JSP、`Servlet` 和 `JavaBean` 技术来实现简单的网上购物功能。读者在实现上述功能的过程中应当仔细体会 JSP、`Servlet` 和 `JavaBean` 各自的作用。

## 第七章 JSP 访问数据库

通过前几章的实例练习，读者应该已经掌握了怎样结合使用 JSP、`Servlet` 和 `JavaBean` 技术实现购物车功能。但是实际应用中的购物车是和数据库技术分不开的，通常所有的商品信息都会从数据库中取出，而不是以隐藏字段的方式存储在 HTML 页面中（参见第四章、第五章和第六章的习题答案，所有的图书信息都是以隐藏字段的形式存储在 HTML 页面中的）。

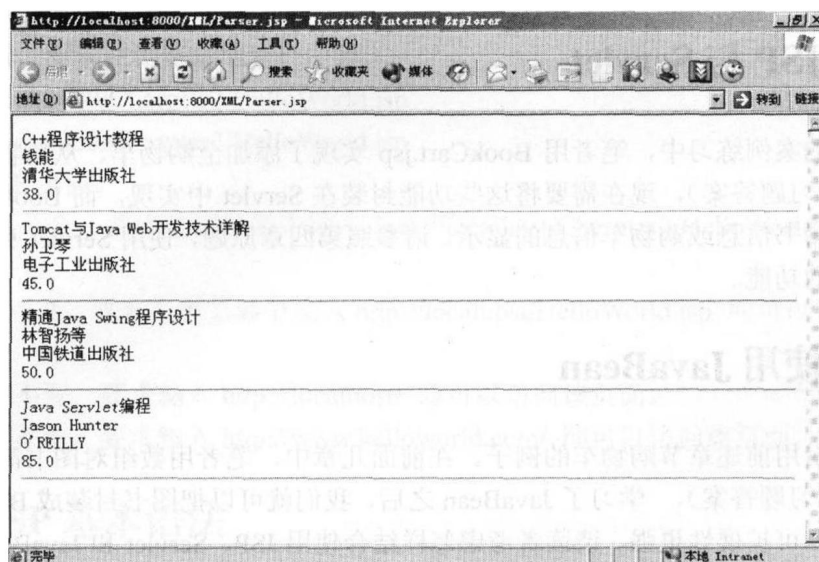
本题要求读者实现这一功能。即在前几章实现购物车功能的基础上，将图书通过数据库存取的方式显示。读者可以参考本书第六章使用 `JavaBean` 封装数据库连接的实例以及第七章的 JSP 访问数据库实例。

## 第八章 JSP 与 XML

1. 编写一个 XML 文件，对第四章案例练习中提供的图书信息进行描述，要求将图书信息以属性（Attribute）形式保存。

2. 写一个 JSP 程序解析上述 XML 文件，可以在浏览器中输出所有的图书信息。结果大致如题图 8.1 所示。

## JSP 开发技术 ( 指导手册 )



题图 8.1

## 第二部分 参考答案

### 第一章 JSP 简介

1. 登录 Chinaren 校友录，网址：<http://alumni.chinaren.com>。这是一个以 JSP 技术为主的网站，有以下一些方法可供你证明这一点：

- ◆ 最简单的证明方法是查看浏览器地址栏的网页后缀名，如果是以 `jsp`（或 `JSP`）结尾，则无疑该网站采用了 JSP 技术。

- ◆ 如果你所登录的网站并没有显示任何网页后缀名（就像 Chinaren 校友录的情况），那么你可以查看网页中所包含的各类链接。当你把鼠标停在带有链接的网页文字或按钮上时，一般网页底部的状态栏就会显示该链接所指向的 HTTP 地址，如果该地址指向的是本网站的另一个 JSP 页面，则该网站必然采用了 JSP 动态网页技术。当然，如果该链接指向的是另外一个网站，则该链接不足以得出任何结论。

- ◆ 查看当前页的 HTML 源代码。在源代码中搜索 `.jsp`（或 `JSP`）关键字，你可以更加清楚地了解到该网站究竟有哪些部分使用了 JSP 技术，或者是根本没有采用任何动态网页技术。

- ◆ 通过 Web 服务器来确定网站所采用的动态网页技术。例如你直接在地址栏中输入：

<http://alumni.chinaren.com/default.jsp>

Web 服务器会报找不到网页的 HTTP 404 错误，同时也会返回 Web 服务器本身的一些信息。Chinaren 校友录的反馈信息如图 1.1 所示。

这表明了 Chinaren 校友录采用了 Resin 1.2.8 作为 Web 服务器，而绝大多数 Web 服务器都和某种动态网页技术关联在一起（例如 IIS 作为 Web 服务器通常会使用 ASP 或 `asp.net` 动态网页技术、而 Apache 通常用作 JSP 或 PHP 动态网页技术的 Web 服务器）。读者如果对此不是很熟悉，可以查询一下 Resin 相关的资料，你会立刻明白使用 Resin 作为 Web 服务器意味着什么。

那么采用同样的方法，你也应该可以知道 Chinaren 校友录是否还采用了其他动态网页技术。

2. 查看 <http://alumni.chinaren.com/index.jsp> 的源文件很简单，选择浏览器“查看”菜单，然后选择“源文件”就可以查看到当前页面的源代码（不同性质的浏览器菜单名可能有所不同）。或者直接点在网页空白处点击鼠标右键，在弹出的快捷菜单中选择“查看源文件”即可。

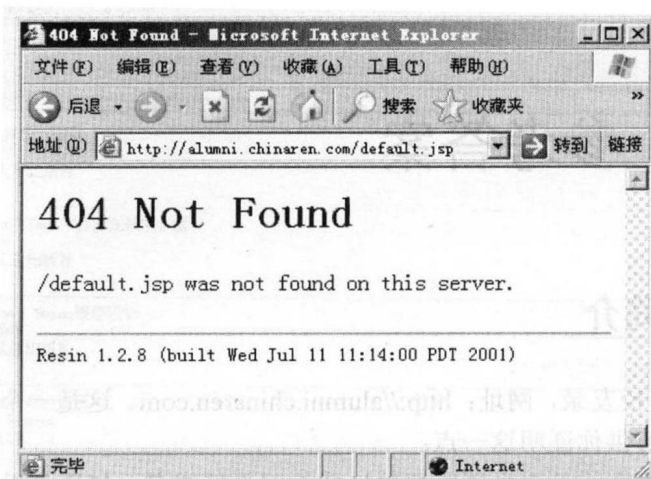


图 1.1

但是你所看到的源代码并非是原始的 JSP 源代码，而是经过 Web 服务器解释之后的静态 HTML 源文件。虽然你访问的是一个 JSP 页面，但返回到你所在的客户端浏览器中的已经是 HTML 静态页面，任何动态网页技术都是如此。因为本质上浏览器只能识别 HTML 文件，这一点读者必须记住。

## 第二章 配置 JSP 运行环境

1. HelloWorld.java 源程序如下：

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

点击 JCreator 工具栏的编译(Compile File)按钮，Debug 栏提示“Process Completed”表示已经成功编译，然后点击工具栏的执行(Execute File)按钮，结果如图 2.1 所示。

2. 在 Tomcat webapps/ROOT 目录下新建一个 Chapter2 目录，在该目录下新建 HelloWorld.jsp 文件，其源代码如下：

```
<%
    out.println("HelloWorld");
```

%>



图 2.1

启动 Tomcat 后，在浏览器中输入 <http://localhost:8000/Charpter2/HelloWorld.jsp>(或 <http://localhost:8080/Charpter2/HelloWorld.jsp>)即可以访问该 JSP 页面。显示结果如图 2.2 所示。

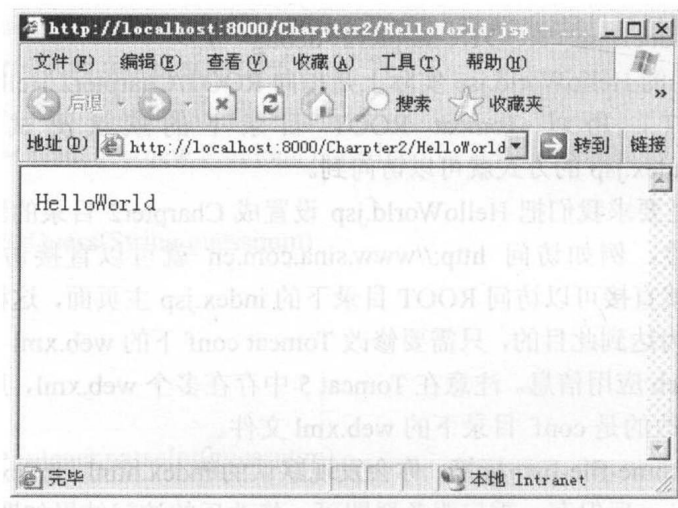


图 2.2

3. 第 3 题和第 2 题不同的是端口号 8000 被取消了，这意味着 HTTP 端口号已经由 8000 改成了默认值 80。那么我们只要把 Tomcat HTTP 访问端口修改为 80 就可以达到目的。Tomcat 中修改 HTTP 访问端口主要有以下两种方法：

◆ 在安装 Tomcat 时直接选择端口为 80, 这样就默认了在访问 Tomcat 服务器时不需要输入端口号。

◆ 修改 Tomcat conf 目录下的 server.xml 文件, 这也是最常用的方法。进入到 Tomcat conf 目录下你会看到一个 server.xml 文件, 该文件是用来存放 Tomcat 服务器各种配置的文件。打开 server.xml, 找到其中的<Connector>元素, 如下所示:

```
<!-- Define a non-SSL Coyote HTTP/1.1 Connector on the port specified
      during installation -->
<Connector port="8000"    maxThreads="150"
      minSpareThreads="25"    maxSpareThreads="75"
      enableLookups="false" redirectPort="8443" acceptCount="100"
      debug="0" connectionTimeout="20000"
      disableUploadTimeout="true" />
```

将其中的 port 属性直接修改成 “80” 然后保存该文件, 重启服务器之后就可以了。

4. 比较第 4 题和第 3 题, 你会发现不需要使用 Charpter2 目录就可以直接访问 Charpter2 目录下的文件。为达到此目的, 我们仍然需要修改 server.xml。观察 server.xml, 在 <HOST></HOST> 标签对之间包括了大量的服务器配置信息, 其中有一个 <Context> 标签, 它代表了运行在 <HOST> 上的一个应用, <Context> 标签的 path 属性和 docBase 属性共同表明了你可以通过 path 的 URL 访问形式访问 docBase 所指定的相对路径。例如, 我们添加以下一行内容:

```
<Context path="" docBase="ROOT/Charpter2" debug="0"/>
```

这表明 URL 中所访问的 HelloWorld.jsp 实际上是指向 ROOT/Charpter2/HelloWorld.jsp。docBase 的默认值是 ROOT, 所以 Tomcat ROOT 目录下的默认测试页 index.jsp 通过 http://localhost:8000/index.jsp 的方式就可以访问到。

5. 第 5 题实际是要求我们把 HelloWorld.jsp 设置成 Charpter2 目录的默认访问页。读者应该有这样的使用经验, 例如访问 http://www.sina.com.cn 就可以直接访问其主页面, 访问 http://localhost:8000 就直接可以访问 ROOT 目录下的 index.jsp 主页面, 这样就省去了网站浏览者输入页面的步骤。为达到此目的, 只需要修改 Tomcat conf 下的 web.xml 文件。该文件通常用来描述已经发布的 Web 应用信息。注意在 Tomcat 5 中存在多个 web.xml, 用以描述不同类型的 Web 应用, 你需要修改的是 conf 目录下的 web.xml 文件。

找到其中的 <welcome-file-list> 标签, 你会发现默认 index.html、index.htm、index.jsp 等页面, 添加 HelloWorld.jsp 后保存, 重启服务器即可。修改后的访问结果如图 2.3 所示。

6. 第 6 题中的 URL 形式是我们最常见的, 实现这个目的需要两步: 首先是能够申请到 www.helloworld.com 域名, 在 DNS 服务器中合法注册, 其次就是修改 Tomcat conf 目录下 server.xml 文件 <HOST> 标签的 name 属性, 设置其值为 www.helloworld.com 即可达到目的。

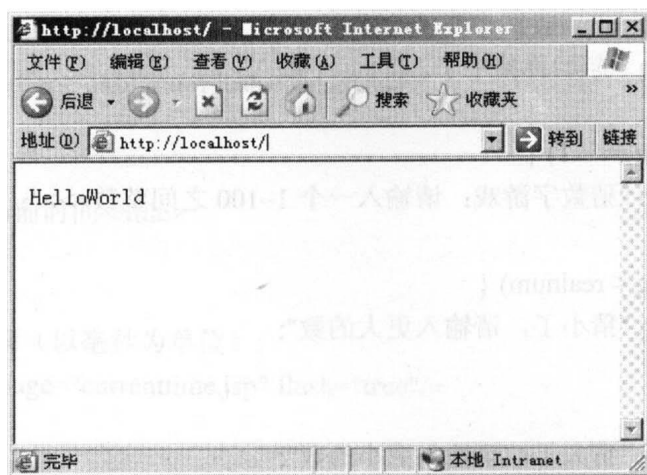


图 2.3

### 第三章 JSP 基本语法

1. 在 EditPlus 或其他你所喜欢的开发工具中输入以下代码，保存为 numGuess.jsp，转移至 Tomcat ROOT 目录下，然后在浏览器中输入 `http://localhost:8000/unmGuess.jsp` 运行测试效果。

numGuess.jsp

```
<%@ page language="java" contentType="text/html; charset=GB2312" %>
<%@ page import="java.util.*" session="false" %>
<%--产生一个 1~100 的随机整数--%>
<%! int realnum = Math.abs(new Random().nextInt() % 100) + 1; %>
<%!
    public String getGuess(String guessnum)
    {
        int g;
        try {

            g = Integer.parseInt(guessnum);
        }
        catch (NumberFormatException e) {
            g = -1;
        }
    }
}
```

```
        if (g == realnum) {
            return "恭喜!你猜中了。";
        }
        else if (g == -1) {
            return "猜数字游戏: 请输入一个 1~100 之间整数>>>";
        }
        else if (g < realnum) {
            return "猜小了, 请输入更大的数";
        }
        else {
            return "猜大了, 请输入更小的数";
        }
    }

    %>
    <%
        String yourguess = (String)request.getParameter("yourguess");
        yourguess = getGuess(yourguess);
        out.println(yourguess);
    %>
    <html>
    <head>
    <title>猜数字游戏</title>
    </head>
    <body>
    <form method=get>
        输入一个数字: <input name="yourguess">
    <input type=submit value="提 交">
    </form>
    </body>
    </html>
```

运行结果如图 3.1 所示。

2. main.jsp 和 currenttime.jsp 源代码分别如下所示:

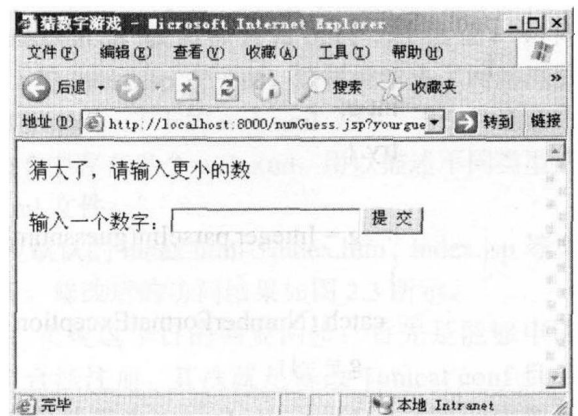


图 3.1

main.jsp

```
<%@ page buffer="5kb" autoFlush="false" %>
<html>
<head>
<title>系统当前时间</title>
</head>
<body>
系统当前时间（以毫秒为单位）：
<jsp:include page="currenttime.jsp" flush="true"/>
</body>
</html>
```

currenttime.jsp

```
<font color="red">
<%= System.currentTimeMillis() %>
</font>
```

运行结果如图 3.2 所示。

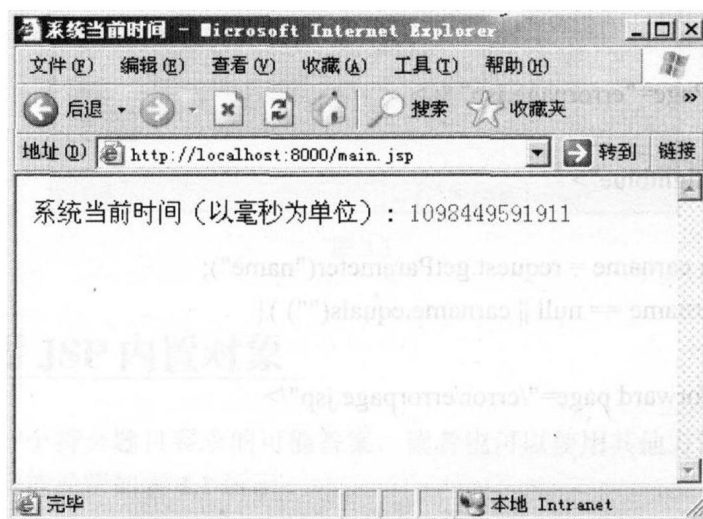


图 3.2

3. 这只是解决问题的其中一个方法，供读者参考。

error.html 供用户选择其喜欢的车型，err.jsp 用来显示用户所选车型，errorpage.jsp 用来处理选择异常。

error.html

```
<html>
```

```
<body>
<h1> 使用 JSP 异常处理对象</h1>
<br>
<h3> 选择我最喜欢的车.</h3>
<form method=get action=err.jsp>
<select name="name">
<option value=""> 选择我最喜欢的车 <br>
<option value="桑塔纳"> 桑塔纳 <br>
<option value="宝马"> 宝马 <br>
<option value="奥迪"> 奥迪 <br>
<option value="红旗"> 红旗 <br>
<option value="帕萨特"> 帕萨特 <br>
</select>
<br> <input type=submit name=submit Value="提 交">
</form>
</body>
</html>
```

err.jsp

```
<%@ page errorPage="errorpage.jsp" %>
<html>
<body bgcolor="lightblue">
    <%
        String carname = request.getParameter("name");
        if ( carname == null || carname.equals("") ) {
    %>
        <jsp:forward page="/error/errorpage.jsp"/>
    <%
        }
        else
            out.println("<H1> 你最喜欢的车是:"+carname+"</H1>");
    %>
</body>
</html>
```

errorpage.jsp

```
<%@ page isErrorPage="true" %>
```

```
<html>
<head>
<title></title>
</head>
<body>
    <h2 color=red> 错误！您没有作出任何选择 </h2>
    <h2 color=red> <%= exception.getMessage() %> </h2>
</body>
</html>
```

运行结果如图 3.3 所示。

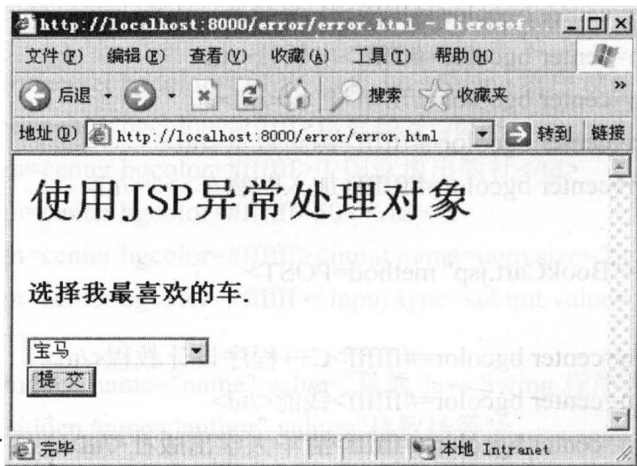


图 3.3

第四章 使用 JSP 内置对象

以下内容只是一个符合题目要求的可能答案，读者也可以使用其他方法实现本题目的。所使用的网页名称及功能说明如表 4.1 所示。

表 4.1

| 网 页 名 称      | 功 能 说 明                             | 网页在 Tomcat 中的位置 |
|--------------|-------------------------------------|-----------------|
| Book.html    | 该网页用以存储和显示可供购买的图书信息，表单可以向购物车提交所购图书。 | ROOT\BookCart\  |
| BookCart.jsp | 实现所购图书添加至购物车、从购物车中删除以及提供结账的链接。      | ROOT\BookCart\  |
| Check.jsp    | 实现结账以及所购图书的显示。                      | ROOT\BookCart\  |

各网页源代码如下:

Book.html

```
<html>
<head>
<title></title>
</head>
<body>
<table cellspacing=1 bgcolor=#000000>
  <tr>
    <td align=center bgcolor=#fffff>书名</td>
    <td align=center bgcolor=#fffff>作者</td>
    <td align=center bgcolor=#fffff>出版社</td>
    <td align=center bgcolor=#fffff>售价</td>
    <td align=center bgcolor=#fffff>购买数量</td>
    <td align=center bgcolor=#fffff>加入购物车>>></td>
  </tr>
  <form action="BookCart.jsp" method=POST>
    <tr>
      <td align=center bgcolor=#fffff>C++程序设计教程</td>
      <td align=center bgcolor=#fffff>钱能</td>
      <td align=center bgcolor=#fffff>清华大学出版社</td>
      <td align=center bgcolor=#fffff>38.0</td>
      <td align=center bgcolor=#fffff><input name=num size=2 value=1></td>
      <td align=center bgcolor=#fffff><input type=submit value="加入购物车"></td>
    </tr>
    <input type=hidden name="name" value="C++程序设计教程">
    <input type=hidden name="author" value="钱能">
    <input type=hidden name="publisher" value="清华大学出版社">
    <input type=hidden name="price" value="38.0">
  </form>
  <form action="BookCart.jsp" method=POST>
    <tr>
      <td align=center bgcolor=#fffff>Tomcat 与 Java Web 开发技术详解</td>
      <td align=center bgcolor=#fffff>孙卫琴</td>
```