软件开发环境

主讲教师 刘凡

fanliu@hhu.edu.cn

第四章

JSP基本语法

本章主要内容

- 4.1 JSP页面的基本结构
- 4.2 变量和方法的声明
- 4.3 Java程序片
- 4.4 Java表达式
- 4.5 JSP中的注释
- 4.6 JSP指令标记
- 4.7 JSP动作标记
- 4.8 小结

4.1 JSP页面基本结构

JSP页面可由5种元素组合而成：

① 普通的HTML标记符

② JSP标记，如指令标记、动作标记

③ 变量和方法的声明

④ Java程序片

⑤ Java表达式

4.1 JSP页面基本结构

- JSP页面中普通的HTML标记符号，交给客户的浏览器执行显示。
- JSP标记、变量和方法声明、Java程序片由Tomcat服务器负责执行，将需要显示的结果发送给客户的浏览器。
- Java表达式由Tomcat服务器负责计算，将结果转化为字符串，交给客户的浏览器负责显示。

4.1 JSP页面基本结构

example2_1.jsp

```
<%@ page contentType="text/html;charset=GB2312"
   %>  <%-- jsp指令标记 -->
<%@ page import="java.util.Date" %><%-- jsp指令标记 -->
<%!   Date date;                                  // 数据声明
     int start,end,sum;
     public int continueSum(int start,int end)  // 方法声明
     {  for(int i=start;i<=end;i++)
           sum=sum+i;
       return sum;
     }
 %>
```

4.1 JSP页面基本结构

```
<HTML><BODY background='back.jpg'>          <!--html标记 -->
<FONT size=4><P>程序片创建Date对象：
  <% date=new Date();                       //java程序片
    out.println("<BR>"+date);
    start=1;
    end=100;
    sum=continueSum(start,end);
  %>
<BR>从
  <%= start %>                          <!-- Java表达式 -->
  至
  <%= end %>
  的连续和是
  <%= sum %>
</FONT></BODY></HTML>
```

4.2 变量和方法的声明

声明变量

在“<%!”和“%>”标记符之间声明变量，变量的类型可以是Java语言允许的任何数据类型，将这些变量称为JSP页面的成员变量。例如：

```
<%!  int  a, b=10 , c;
        String tom=null,jerry="love JSP";
        Date date;
 %>
```

4.2 变量和方法的声明

example2_2.jsp：利用成员变量被所有用户共享，实现计数器

```
<%@ page contentType="text/html; charset=GB2312"
   %>
<HTML><BODY BGCOLOR=cyan>
<FONT size=3>
 <%! int i=0; %>
 <% i++; %>
<P>您是第
  <%= i %>
 个访问本站的客户。
</BODY></HTML>
```

4.2 变量和方法的声明

声明方法

- 在“<%!”和“%>”标记符号之间定义方法，所定义的方法在整个JSP页面有效，可以在Java程序片中被调用。

```
<%!
    String getDate(){
      String tom;
      tom= java.util.Date().toLocalString();
      return tom;
    }
%>
```

4.2 变量和方法的声明

example2_2.jsp

```
<%@ page contentType="text/html;charset=GB2312" %>
<HTML><body bgcolor=#FFAAEE>
    <%! double multi(double x,double y){
            return x*y;
        }
        double div(double x,double y) {
            return x/y;
        }
    %>
    <%  double x=3.56;
        double y=18;
        out.println("调用multi方法计算"+x+"与"+y+"之积：<br>");
        out.println(multi(x,y));
        out.println("<BR>调用div方法计算"+y+"除以"+x+"的商：<br>");
        String s =String.format("小数点保留3位：%10.3f",div(y,x));
        out.println(s);
    %>
</body></HTML>
```

4.3 Java程序片

- 在“<%”和“%>”之间插入Java程序片（合法的Java代码）
- 程序片中声明的变量称为JSP页面的局部变量，在JSP页面后继的所有程序片和表达式内有效。
- 多个客户请求一个JSP页面时，Java程序片将被执行多次，分别在不同的线程中执行，互不干扰。

4.3 Java程序片

- 因为**JSP**页面实际上是被编译成**Servlet**类执行的，所以声明中定义的变量是**Servlet**类的成员变量，各个用户共享成员变量，需同步。
- 程序片中定义的变量是局部变量，用户之间没有联系，每次调用页面，局部变量都被重新初始化。

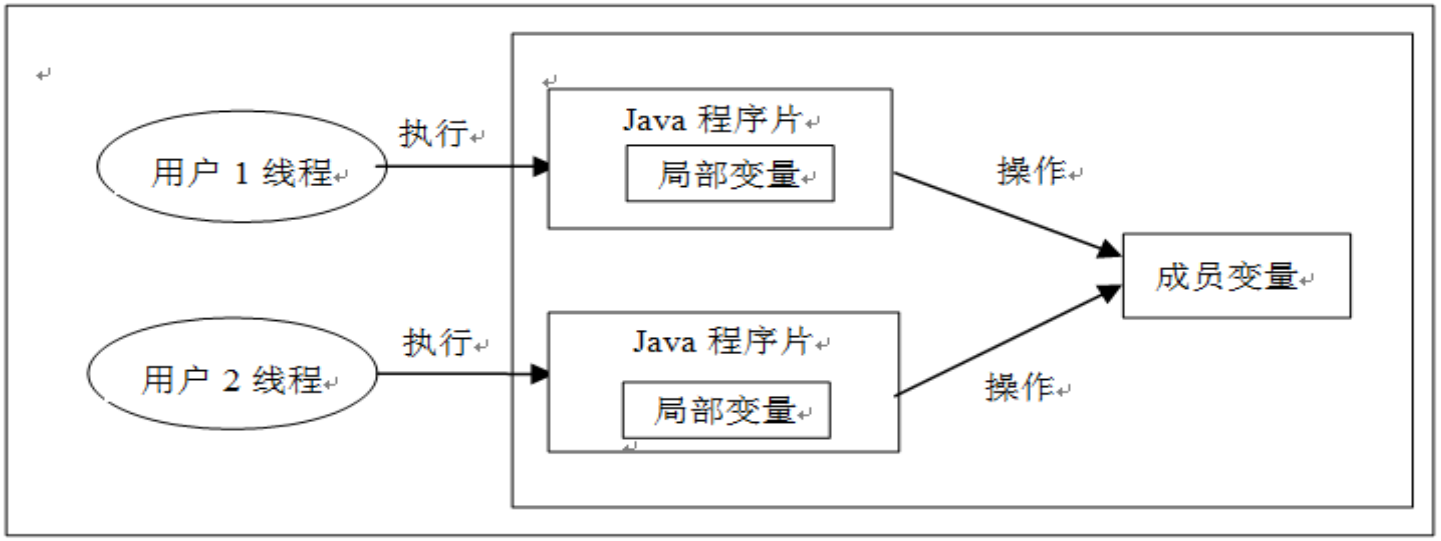

图 2.4 程序片的执行

4.3 Java程序片

example2_4.jsp

```
<%@ page contentType="text/html;Charset=gb2312" %>
<HTML><BODY>
   <%! int count=0;                    //被用户共享的count
   synchronized void setCount()  //synchronized修饰的方法
      {   count++;
      }
   %>
   <%  setCount();
       out.println("您是第"+count+"个访问本站的用户");
   %>
</BODY></HTML>
```

4.3 Java程序片

- 一个JSP页面中的Java程序片会按其在页面中的顺序被执行，而且某个Java程序片中声明的局部变量在其后继的所有Java程序片以及表达式内都有效。
- 可以将一个Java程序片分割成几个Java程序片，然后在这些Java程序片之间再插入其他标记元素。

```
<%
    Java程序片1
%>
    HTML标记
<%
    Java程序片2
%>
    HTML标记
```

4.3 Java程序片

example2_5.jsp

```
<%@ page contentType="text/html;charset=GB2312" %>
<HTML><body bgcolor=cyan>
  <%  //Math.random()是（0,1）之间的随机数
      int number = 7+(int)(Math.random()*13);
      if(number<=13) {
  %>   <center> <h2> 显示小学生图片</h2> <!-- 插入其他标记 -->
    <image src ='image/xiao.jpg' width=180 height=178>小学生</image>
  <%  }
      else {
  %>   <center> <h2> 显示中学生图片</h2>
<image src ='image/zhong.jpg' width=180 height=178>中学生</image>
  <%  }
  %>
</body></HTML>
```

4.4 Java表达式

- 可以在“<%=”和“%>”之间插入一个表达式，这个表达式必须能求值。表达式的值由服务器负责计算，并将计算结果用字符串形式发送到用户端显示。

  <%=getDate()%>

- 注意：不可插入语句，“<%=”是一个完整的符号，“<%”和“=”之间不要有空格.

4.4 Java表达式

example2_6.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
<HTML><body  bgcolor=cyan><font size=3>
<%  int x=12,y=9;
%>
  计算表达式x+y+x%y,即
<%=x%>+<%=y%>+<%=x%>%<%=y%>的值:
  <%= x+y+x%y %>
<br>计算表达式x>y即<%=x%> > <%=y%>的值:
  <%= x>y %>
<br>计算表达式sin(<%=Math.PI %>/2)的值:
    <%= Math.sin(Math.PI/2)%>
<br>
<%  if(x-y>=0) {
%>     如果<%=x%>大于<%=y%>计算<%=x%>与
<%=y%>
       的差:<%=x-y%>的平方根:
       <%=Math.sqrt(x-y)%>
<%  }
%>
</font></body></HTML>
```

4.5 JSP中的注释

- **HTML注释格式:**

  <!--　注释内容　-->　显示注释

- **JSP注释格式:**

  <%--　注释内容　--%>　隐式注释

  JSP注释写在**JSP**程序中，但不发送给客户。

- **Scriptlets中的注释:**

由于**Scriptlets**包含的是**java**代码，所以**java**中的注释规则在**scriptlets**中也适用，常用的**java**注释使用**//**表示单行注释，使用**/*　*/**表示多行注释. 隐式注释

4.5 JSP注释

example2_7.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
<HTML><body>
 <!-- 以下字体的颜色为蓝色 -->
 <FONT size=3 color=blue>抽取字符串"C:\myfile\jspfile\example.jsp"中的"example.jsp"
</FONT>
  <%-- 下面是成员变量的声明 --%>
 <%!  String s="C:\\myfile\\jspfile\\example.jsp";
 %>
 <%-- 下面是Java程序片 --%>
 <%  int index=s.lastIndexOf("\\");
     String str=s.substring(index+1);
 %>
 <BR><%-- 下面是Java表达式 --%>
 <%= str %>
</body></HTML>
```

4.6 JSP指令标记

JSP中主要有2种指令标记：

- page指令标记
- include指令标记

4.6 JSP指令标记

page指令标记:

- page 指令用来定义整个JSP页面的一些属性和这些属性的值，属性值用单引号或双引号括起来。
- page指令与其书写的位置无关，习惯把page指令写在JSP页面的最前面。
- <%@ page 属性1=“属性1的值” 属性2= “属性2的值” ……%> 或

  <%@ page 属性1=″属性1的值″ %>

  <%@ page 属性2=″属性2的值″ %>

4.6 JSP指令标记

page指令标记：contentType属性

- contentType 属性值确定JSP页面响应的MIME类型和JSP页面字符的编码。属性值的一般形式是"MIME类型”或 "MIME类型;charset=编码"

<%@ page contentType=“application/msword” %>

- 如果不使用page指令为contentType指定一个值，那么contentType默认值是 "text/html ; charset=ISO-8859-1 "

4.6 JSP指令标记

- 注：不允许两次使用page 指令给contentType属性指定不同的属性值 .
- 例子2_8中example2_8.jsp页面使用page指令设置contentType属性的值是"image/x-xbitmap"，当用户请求example2_8.jsp页面时，用户的浏览器将启用图形解码器来解析执行收到的信息。

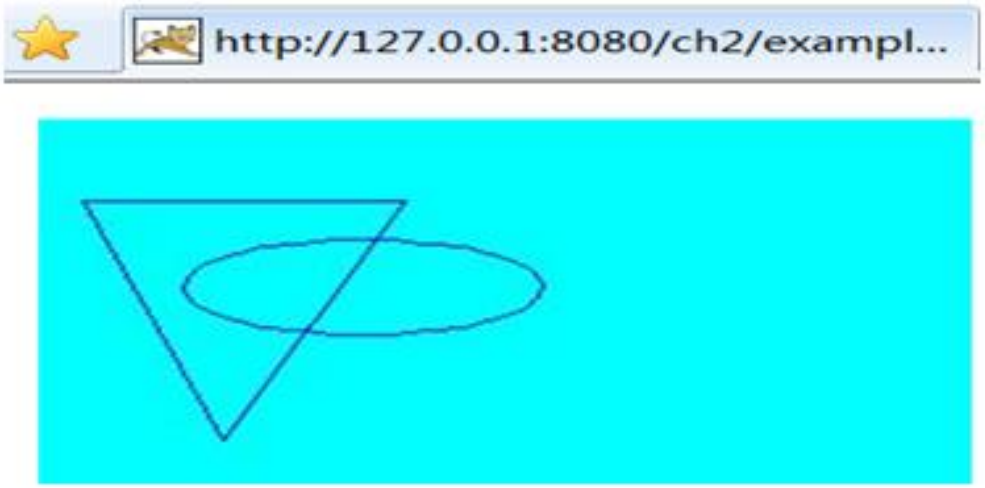

图 2.7 contentType 的值是"image/x-xbitmap"

4.6 JSP指令标记

page指令标记：language属性

- language属性定义JSP页面使用的脚本语言，该属性的值目前只能取“java”。
- 例如:

```
<%@ page  language="java" %>
```

- 注：JSP页面默认有如上page指令 。

4.6 JSP指令标记

page指令标记：import属性

- 该属性的作用是为JSP页面引入Java运行环境提供的包中的类，这样就可以在JSP页面的程序片部分、变量及函数声明部分、表达式部分使用包中的类。
- **<%@ page import="java.io.*", "java.util.Date" %>**
- 注：JSP页面默认import属性已经有" java.lang.*"、"javax.servlet.*"、"javax.servlet.jsp.*"、"javax.servlet.http.*"等值 。

4.6 JSP指令标记

page指令标记：session属性

- session 属性用于设置是否需要使用内置的session对象。
- session的属性值可以是true或false。session属性默认的属性值是true。

4.6 JSP指令标记

page指令标记：buffer属性

- 内置输出流对象out负责将服务器的某些信息或运行结果发送到用户端显示。buffer属性用来指定out设置的缓冲区的大小或不使用缓冲区。例如：

```
<%@ page buffer= "24kb" %>
```

- buffer属性的默认值是8kb 。buffer属性可以取值" none"，即设置out不使用缓冲区。

4.6 JSP指令标记

page指令标记：autoFlush属性

- autoFlush 属性指定out的缓冲区被填满时，缓冲区是否自动刷新。
- autoFlush属性的默认值是true。
- 当autoFlush属性取值false时，如果out的缓冲区填满，就会出现缓存溢出异常。当buffer的值是“none”时，autoFlush的值就不能设置成false。

4.6 JSP指令标记

page指令标记： isThreadSafe属性

- isThreadSafe属性用来设置JSP页面是否可多线程访问。
- 当isThreadSafe属性值设置为true时，JSP页面能同时响应多个用户的请求；当isThreadSafe属性值设置成false时，JSP页面同一时刻只能响应一个用户的请求，其他用户须排队等待
- 注： isThreadSafe属性的默认值是true。

4.6 JSP指令标记

page指令标记： info属性

- info属性的属性值是一个字符串，其目的是为JSP页面准备一个常用且可能要经常修改的字符串。例如，

```
<%@ page info= "we are students" %>
```

- 可以在JSP页面中使用方法：getServletInfo();获取info属性的属性值。
- 注意：当JSP页面被转译成Java文件时，转译成的类是Servlet的一个子类，所以在JSP页面中可以使用Servlet类的方法：getServletInfo()。

4.6 JSP指令标记

example2_9.jsp

```
<%@ page
contentType="text/html;charset=gb2312" %>
<%@ page info="清华大学图像tsinghua.jpg" %>
<%  String s=getServletInfo();
   String str[]=s.split("图像");
%>
<HTML><center>
<body background="image/<%=str[1]%>"><font
Size=4>
<br><%=str[0] %>出版社是中国著名出版社
<br><%=str[0] %>是全国著名的高等学府
</font></body></center><HTML>
```

4.6 JSP指令标记

include指令标记:

- include指令标记的作用是在JSP页面出现该指令的位置处，静态插入一个文件，实现代码的复用。

<%@ include file= "文件的URL " %>

- 一个 JSP 页面中的 include 指令的数量不受限制。
- 静态插入，就是当前JSP页面和插入的文件合并成一个新的JSP页面，然后JSP引擎再将这个新的JSP页面转译成Java文件。

4.6 JSP指令标记

```
<html>
  <head>
    <title>Include directive test
page </title>
  </head>
  <body>
    <h1>Include directive test
</h1>

    <%@ include
file="/copyright.html" %>

  </body>
    </html>
```

嵌入

copyright.html：

```
<p>&copy; 2002 Microsoft
 Corp.</p>
```

4.6 JSP指令标记

include指令标记:

- 使用include指令可以把一个复杂的JSP页面分成若干简单的部分，当要对页面更改时，只需更改对应的部分就行了。
- 通常情况下把页面分成多个区，典型的分区方法如下：

<table>
<tr><td colspan="2">头部：head.jsp，LOGO</td></tr>
<tr><td>左边：
side.jsp，
菜单</td><td>页面主体：
body.jsp，
功能区</td></tr>
<tr><td colspan="2">尾部：footer.jsp，版权声明等</td></tr>
</table>

4.6 JSP指令标记

例子2_10中，两个JSP页面example2_10_a.jsp, example2_10_b.jsp使用include指令标记嵌入相同的一个文本文件ok.txt，实现这两个JSP页面之间的超链接。

ok.txt

```
<%@ page
contentType="text/html;charset=gb2312" %>
<center>
<A href="example2_10_a.jsp">北京大学</A>
<A href="example2_10_b.jsp">清华大学</A>
```

4.7 JSP动作标记

JSP规范定义了一系列的标准动作，它们都以jsp为前缀，常用的有：

<jsp:param> <jsp:expression>

<jsp:include> <jsp:attribute> <jsp:plugin>

<jsp:forward> <jsp:body> <jsp:invoke>

<jsp:useBean> <jsp:text> <jsp:output>

<jsp:setProperty> <jsp:element> <jsp:doBody>

<jsp:getProperty> <jsp:scriptlet> <jsp:root>

4.7 JSP动作标记

include动作标记:

- include动作标记告诉JSP页面动态包含一个文件，即JSP页面运行时才将文件加入。

```
<jsp:include page= "文件的URL"/>
```

或

```
<jsp:include page= "文件的URL">
    param子标记
</jsp:include>
```

- 如果是普通的文本文件，就将文件的内容发送到用户端，由浏览器负责显示；如果是JSP文件，JSP引擎就执行这个文件，然后将执行的结果发送到用户端浏览器显示。

4.7 JSP动作标记

param动作标记:

- param标记以“名字—值”对的形式为其他标记提供附加信息。
- param 动作标记语法格式为

```
<jsp:param  name= "名字"  value= "指定给param的值">
```

- 注意： param 标记不能独立使用，需作为 jsp:include 、jsp:forward、jsp:plugin标记的子标记来使用

4.7 JSP动作标记

param动作标记:

- 当该标记与jsp:include动作标记一起使用时，可以将param标记中的值传递到include动作标记要加载的文件中去，被加载的JSP文件可以使用Tomcat服务器提供的request内置对象获取include动作标记的param子标记中name属性所提供的值。

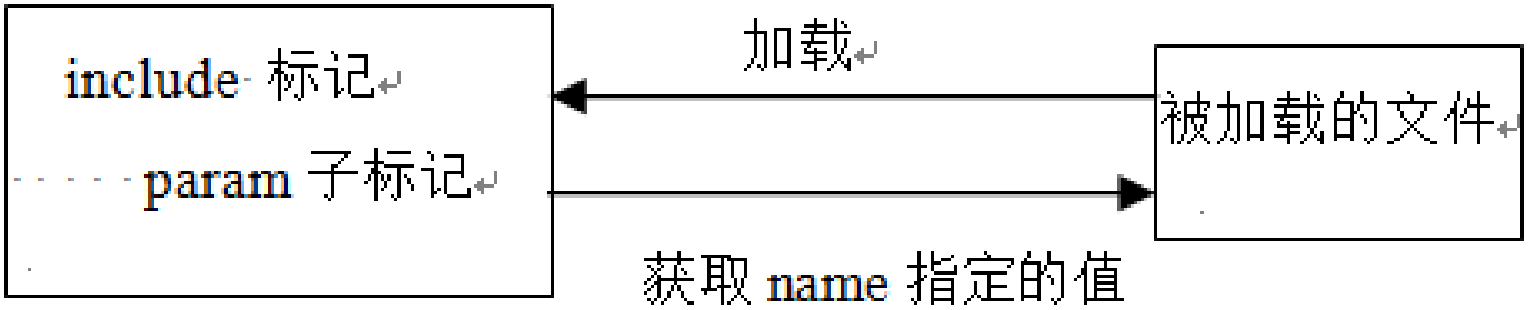

图 2.10 使用 include 动作标记加载文件

4.6 JSP指令标记

example2_11.jsp

```
<%@ page
contentType="text/html;charset=GB2312" %>
<HTML><body  bgcolor=cyan >
<% double a=3,b=4,c=5;
%>
<br>加载trangle.jsp计算三边为
<%=a%>,<%=b%>,<%=c%>的三角形面积.
  <jsp:include page="myfile/trangle.jsp">
   <jsp:param name="sideA" value="<%=a%>"/>
   <jsp:param name="sideB" value="<%=b%>"/>
   <jsp:param name="sideC" value="<%=c%>"/>
  </jsp:include>
</body></HTML>
```

4.6 JSP指令标记

trangle.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
<%! public String getArea(double a,double b,double c) {
      if(a+b>c&&a+c>b&&c+b>a) {
         double p=(a+b+c)/2.0;
         double area=Math.sqrt(p*(p-a)*(p-b)*(p-c)) ;
         return ""+area;
      }
      else { return(""+a+","+b+","+c+"不能构成一个三角形,无法计算面积"); }
   }
 %>
 <%String sideA=request.getParameter("sideA");
     String sideB=request.getParameter("sideB");
     String sideC=request.getParameter("sideC");
     double a=Double.parseDouble(sideA);
     double b=Double.parseDouble(sideB);
     double c=Double.parseDouble(sideC);
 %> <font color=blue size=2>
 <br><b>我是被加载的文件,负责计算三角形的面积<br>
   给我传递的三边是:<%=sideA%>,<%=sideB%>,<%=sideC%></b>
 <br><b><i>三角形的面积:<%= getArea(a,b,c)%></i></b>
 </font>
```

4.7 JSP动作标记

forward动作标记:

```
<jsp:forward page="要转向的页面" />
或 <jsp:forward page="要转向的页面" >
    param子标记
  </jsp:forward>
```

- 该指令的作用是：从该指令处停止当前页面的执行，而转向执行page属性指定的JSP页面。
- 也可以使用param动作标记作为子标记传送信息，要转向的JSP页面用request内置对象获取param子标记中name属性所提供的值。

4.7 JSP动作标记

forward动作标记:

注意：

- 当forward动作标记不需要param子标记时，必须使用第一种形式。
- 当前页面使用forward动作标记转向后，尽管用户看到了转向后的页面的效果，但浏览器地址栏中显示的仍然是转向前的JSP页面的URL地址，因此，如果刷新浏览器的显示，将再次执行当前浏览器地址栏中显示的JSP页面。

4.6 JSP指令标记

example2_12.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
<HTML><body>
<h1> 产生一个1-10之间的随机数
<%  double i=(int)(Math.random()*10)+1;
   if(i<=5) {
%>      <jsp:forward page="example2_12_a.jsp" >
         <jsp:param name="number" value="<%= i %>" />
       </jsp:forward>
<%  }
   else {
 %>    <jsp:forward page="example2_12_b.jsp" >
         <jsp:param name="number" value="<%= i %>" />
       </jsp:forward>
 <% }
 %>
</body></HTML>
```

4.7 JSP动作标记

useBean动作标记:

- 该标记用来创建并使用一个Javabean，是非常重要的一个动作标记，将在第4章详细讨论。
- <jsp:useBean id="id"
scope="page|request|session|application" typeSpec/>
- 其中id表示实例；scope表示此对象可以使用的范围

4.7 JSP动作标记

setProperty动作标记:

- 此操作与useBean协作，用来设置Bean的简单属性和索引属性。<jsp:setProperty>标签使用Bean给定的setXXX()方法，在Bean中设置一个或多个属性值。利用<jsp:setProperty>设置属性值有多种方法。
- <jsp:setProperty name="beanName" propertyDetails/>

4.7 JSP动作标记

getProperty动作标记:

- jsp:getProperty操作是对jsp:setProperty操作的补充，它用来访问一个Bean的属性。
- <jsp:getProperty name="userSession" property="name"/>

4.8 小结

- JSP页面：普通的HTML标记（客户端浏览器执行）、JSP标记、成员变量和方法声明、Java程序片、Java表达式（JSP引擎处理并将结果发送给用户浏览器）。
- 成员变量为所有用户共享，任何用户对成员变量的操作都会影响其他用户，synchronized关键字保证一次只有一个线程执行。
- 多用户访问JSP页面，其程序片会被执行多次，分别在不同线程中，其局部变量互不干扰。

4.8 小结

- page指令标记用来定义整个JSP页面的一些属性，常用的有contentType和import。
- include指令标记在编译阶段就处理所需要的文件，被处理的文件在逻辑与语法上依赖于当前JSP页面，优点是速度快；include动作标记是在JSP页面运行时才处理文件，在逻辑与语法上独立于当前JSP页面，更加灵活。