



软件开发环境

主讲教师 刘凡

fanliu@hhu.edu.cn



第六章

JSP与Javabeen



本章主要内容

- ◆ 6.1 编写Javabean和使用Javabean
- ◆ 6.2 获取和修改beans的属性
- ◆ 6.3 beans的辅助类
- ◆ 6.4 JSP与bean结合的简单例子
- ◆ 6.5 小结

概述

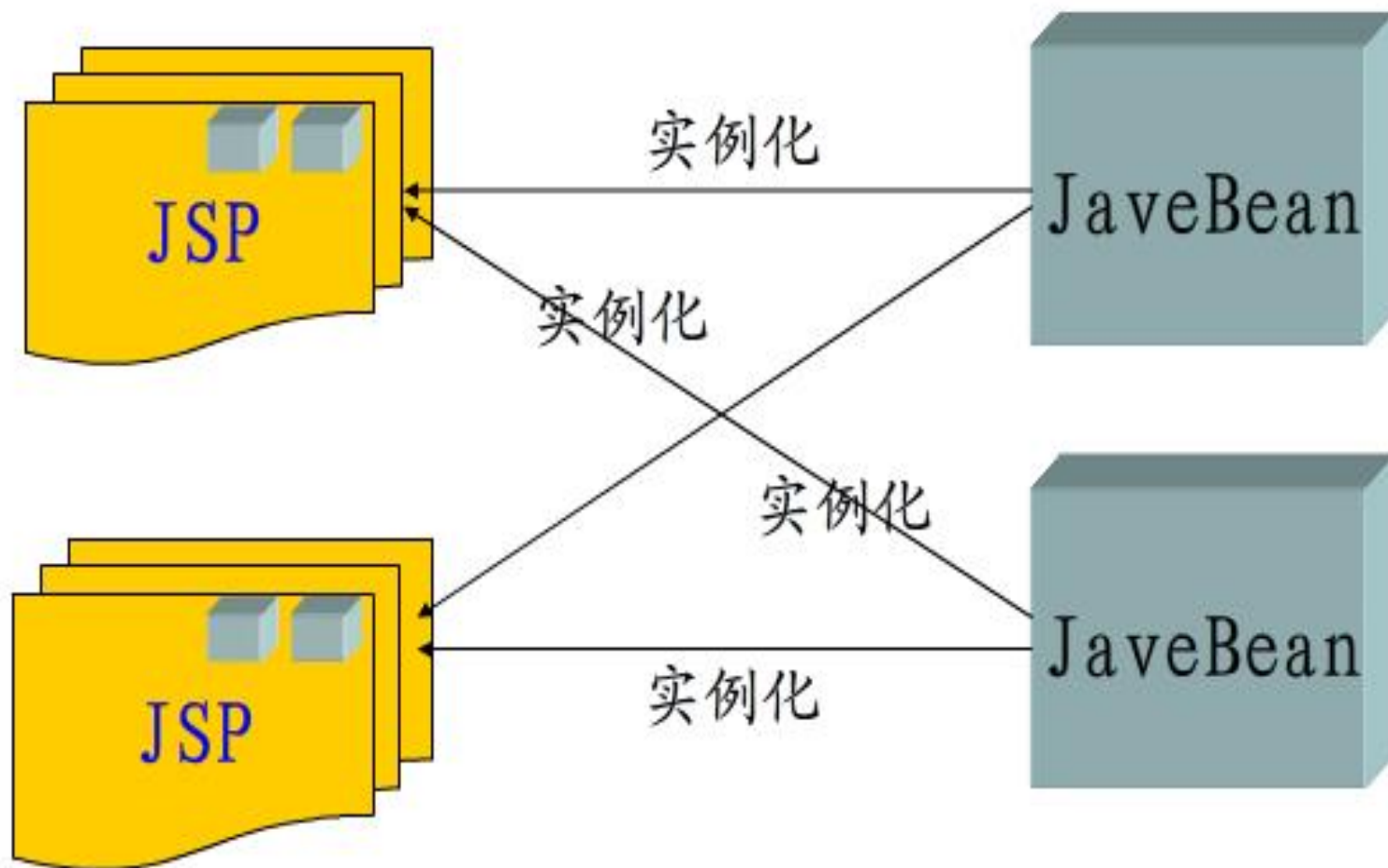
- JavaBean是一个可重复使用的软件组件，是遵循一定标准、用Java语言编写的一个类，该类的一个实例称为一个JavaBean，简称bean。
- 一般实现网页中的业务逻辑或数据库操作。

概述

JavaBean特点:

- 可以实现代码的重复利用
- 易编写、易维护、易使用
- 跨平台

JavaBean工作原理



JSP页面将数据的处理过程指派给一个或几个Bean来完成，
然后负责显示Bean中的数据

6.1 编写Javabeen和使用Javabeen

1. 编写Javabeen

2. 保存Javabeen的字节码

3. 创建和使用Javabeen

6.1 编写Javabeen和使用Javabeen

1.编写Javabeen

- 如果类的成员变量的名字是xxx，那么为了获取或更改成员变量的值，即获取或更改属性，类中必须提供两个方法：

getXxx() 用来获取属性xxx

setXxx() 用来修改属性xxx

- 对于boolean类型的成员变量，即布尔逻辑类型的属性，允许使用“is”代替上面的“get”和“set”。
- 类中声明的方法的访问属性都必须是public的。
- 类中声明的构造方法必须是public、无参数的。

6.1 编写Javabean和使用Javabean

Circle.java

package tom.jiafei;

public class Circle {

double radius;

public Circle() {

radius=1;

}

public double getRadius() {

return radius;

}

public void setRadius(double newRadius) {

radius=newRadius;

}

public double circleArea() {

return Math.PI*radius*radius;

}

public double circlLength() {

return 2.0*Math.PI*radius;

}

}

6.1 编写Javabeen和使用Javabeen

2.Bean字节码的保存

- 在当前Web服务目录下建立如下目录结构:

Web服务目录\WEB-INF\classes

- 根据类的包名，在目录classes下建立相应的子目录. 比如:

Web服务目录\WEB-INF\classes\tom\jiafei

- 将Bean字节码保存在相应的子目录.

Web服务目录\WEB-INF\classes\tom\jiafei

6.1 编写Javabean和使用Javabean

3.使用Javabean

➤ 使用JSP动作标记：useBean，来加载使用bean。useBean标记的语法格式：

■ **<jsp:useBean id="bean的名字" class="创建beans的字节码" scope="bean有效范围"/>**

■ **<jsp:useBean id= "bean的名字" class="创建beans的类" scope= "bean有效范围">
</jsp:useBean>**

➤ 例如：

<jsp:useBean id="circle" class="tom.jiafei.Circle" scope="page" />

6.1 编写Javabean和使用Javabean

4.Bean的加载原理

- JSP引擎加载Bean时首先查找内置pageContext对象中是否存在这样的Bean，如存在则分配给用户，否则根据class指定的字节码文件创建一个Bean，并添加到PageContext对象中。
- **注意：**如果修改了字节码文件，必须重启JSP引擎才能使用。

6.1 编写Javabean和使用Javabean

5.Bean的有效范围（生命周期）

- Scope取值：page、request、session、application
- page：JSP引擎分配给每个JSP页面的bean是互不相同的，不同用户的scope取值是page的bean也是互不相同的。
- request：bean的有效范围是request期间，JSP引擎分配给每个JSP页面的request期间的Bean是互不相同的，不同用户的scope取值是request的bean也是互不相同的。

6.1 编写Javabean和使用Javabean

5.Bean的有效范围（生命周期）

- Scope取值: `page`、`request`、`session`、`application`
- `session`: bean的有效范围是用户的会话期间，不同页面的bean是同一个bean，但不同用户的scope取值为session的bean是互不相同的。
- `application`: JSP引擎为Web服务目录下所有的JSP页面分配一个共享的bean，不同用户的bean是同一个，直到服务器关闭。

6.1 编写Javabean和使用Javabean

example4_1.jsp;

```
<%@ page contentType="text/html;charset=GB2312" %>
<HTML><body bgcolor=cyan><font size=3>
<jsp:useBean id="circle" class="tom.jiafei.Circle" scope="page" />
  <%--通过上述JSP标记，用户获得了一个作用域是page，名字是circle
    的bean --%>
  <P>圆的初始半径是： <%=circle.getRadius()%>
  <% double newRadius =100;
    circle.setRadius(newRadius); //修改半径
  %>
  <P>修改半径为<%= newRadius %>
  <br><b>目前圆的半径是： <%=circle.getRadius()%>
  <br><b>圆的周长是： <%=circle.circlLength()%>
  <br><b>圆的面积是： <%=circle.circleArea()%>
</font></body></HTML>
```

6.1 编写Javabean和使用Javabean

example4_2_a.jsp;

```
<%@ page contentType="text/html;charset=GB2312" %>
```

```
<HTML><body bgcolor=cyan><font size=3>
```

```
<%--通过JSP标记，用户获得一个作用域是session，id是girl的bean: --%>
```

```
<jsp:useBean id="girl" class="tom.jiafei.Circle"scope="session" />
```

```
<br>这里是example4_2_a.jsp页面
```

```
<br>圆的半径是: <%=girl.getRadius()%>
```

```
<a href="example4_2_b.jsp"><br>example4_2_b.jsp </a>
```

```
</font></body></HTML>
```


6.1 编写Javabean和使用Javabean

example4_2_b.jsp;

```
<%@ page contentType="text/html;charset=GB2312" %>
<HTML><body ><font size=3>
<%--通过JSP标记，用户获得一个作用域是session，id是girl的bean: --%>
<jsp:useBean id="girl" class="tom.jiafei.Circle" scope="session" />
  <i><br>这里是example4_2_b.jsp页面
  <br>当前圆的半径是： <%=girl.getRadius()%>
  <% girl.setRadius(99);
  %>
  <br>修改后的圆的半径是： <%=girl.getRadius()%></i>
  <a href="example4_2_a.jsp"><BR>example4_2_a.jsp</a>
</font></body></HTML>
```

6.1 编写Javabean和使用Javabean

example4_3.jsp;

```
<%@ page contentType="text/html;charset=gb2312" %>
```

```
<HTML><body bgcolor=yellow>
```

```
<jsp:useBean id="boy" class="tom.jiafei.Circle" scope="application" />
```

```
圆的初始半径是: <%=boy.getRadius()%>
```

```
<%= boy.setRadius(1000);
```

```
%>
```

```
<br><b>修改后的圆的半径是: <%=boy.getRadius()%>
```

```
</body></HTML>
```

6.2 获取和修改bean的属性

- 获取和修改bean的属性还可以使用动作标记getProperty、setProperty;
- 使用getProperty、setProperty标记获取或修改属性xxx时, 必须保证bean有相应的getXxx和setXxx方法;
- 而在程序片中直接用bean调用方法就不需要方法命名遵守getXxx和setXxx规则。

6.2 获取和修改bean的属性

1.getProperty

- 使用getProperty动作标记可以获得bean的属性值，并将这个值用串的形式发送给用户的浏览器。使用getProperty动作标记之前，必须使用useBean动作标记获得相应的bean。

`<jsp:getProperty name= "bean的名字" property= "bean的属性" />`

或

`<jsp:getProperty name= "bean的名字" property= "bean的属性"/>`

`</jsp:getProperty>`

相当于Java表达式：

`<%= bean.getXxx() %>`

6.2 获取和修改bean的属性

2.setProperty

- 使用setProperty动作标记可以设置bean的属性值, 使用这个标记之前, 必须使用useBean标记得到一个相应的bean。

(1) 将bean属性的值设置为一个表达式的值或字符串

```
<jsp:setProperty name="bean的名字" property="bean的属性" value="<%=expression%>" />
```

```
<jsp:setProperty name=" bean的名字" property="bean的属性" value=字符串 />
```

6.2 获取和修改bean的属性

2.setProperty

(2) 通过HTTP表单的参数值来设置bean的相应属性的值

- 用HTTP表单的所有参数的值设置bean相对应的属性的值，**参数名必须一致**：

```
<jsp:setProperty name= "bean的名字" property="*" />
```

- 用HTTP表单的某个参数的值设置bean的某个属性的值，**不求参数名一致**：

```
<jsp:setProperty name= " bean的名字" property=
"bean属性名" param= "表单中的参数名" />
```

6.2 获取和修改bean的属性

example4_4: Student.java

```
package tom.jiafei;
public class Student {
    String name=null;
    double height;
    public String getName() {
        return name;
    }
    public void setName(String newName){
        name=newName;
    }
    public double getHeight() {
        return height;
    }
    public void setHeight(double newHeight) {
        height=newHeight;
    }
}
```

6.2 获取和修改bean的属性

example4_4.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
<jsp:useBean id="zhang" class="tom.jiafei.Student"
  scope="request" />
<HTML><body bgcolor=pink><font size=3>
<jsp:setProperty name="zhang" property="name" value="张三"
  />
<b> 学生姓名: <jsp:getProperty name="zhang"
  property="name" /></b>
<jsp:setProperty name="zhang" property="height"
  value="<%=1.78%>" />
<br><b> 学生身高: <jsp:getProperty name="zhang"
  property="height" />米</b>
</font></body></HTML>
```


6.2 获取和修改bean的属性

example4_5: Goods.java

```
package tom.jiafei;
public class Goods {
    String name="无名";
    double price=0;
    public String getName() {
        return name;
    }
    public void setName(String newName){
        try{
            byte [] bb = newName.getBytes("iso-8859-1");
            newName = new String(bb);
        }
        catch(Exception exp){}
        name=newName;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double newPrice) {
        price=newPrice;
    }
}
```

6.2 获取和修改bean的属性

example4_5_a.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
<jsp:useBean id="iPhone" class="tom.jiafei.Goods" scope="request" />
<HTML><body bgcolor=yellow><font size=3>
  <form action="" Method="post" >
    手机名称:<input type=text name="name">
    <br>手机价格:<input type=text name="price">
    <br><input type=submit value="提交给本页面">
  </form>
  <form action="example4_5_b.jsp" Method="post" >
    手机名称:<input type=text name="name">
    <br>手机价格:<input type=text name="price">
    <br><input type=submit value="提交给example4_5_b.jsp页面">
  </form>
  <jsp:setProperty name="iPhone" property="name" param ="name" />
  <jsp:setProperty name="iPhone" property="price" param ="price"/>
  <br><b>名称: <jsp:getProperty name="iPhone"
    property="name"/>
  <br><b>价格: <jsp:getProperty name="iPhone"
    property="price"/>
</font></body></HTML>
```

6.2 获取和修改bean的属性

example4_5_b.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
<jsp:useBean id="iPhone" class="tom.jiafei.Goods"
  scope="request" />
<HTML><body bgcolor=cyan><font size=3>
  <jsp:setProperty name="iPhone" property="name" param
    ="name" />
  <jsp:setProperty name="iPhone" property="price" param
    ="price" />
  <br><b>名称: <jsp:getProperty name="iPhone"
    property="name" />
  <br><b>价格: <jsp:getProperty name="iPhone"
    property="price" />
</font></body></HTML>
```

6.3 bean的辅助类

- 有时在写一个bean的时候，除了需要用import语句引入JDK平台提供的类，可能还需要其它自己编写的一些类。
- 那么只要将这些类和创建bean的类写在一个java源文件中即可，但必须将源文件编译后产生的全部字节码文件复制到相应的目录中。

6.3 bean的辅助类

例子6中，使用一个bean列出JSP页面所在目录中特定扩展名的文件。在写bean的类文件**ListFile**时，我们需要一个实现**FilenameFilter**接口的辅助类**FileExtendName**，该类可以帮助bean列出指定扩展名的文件。

ListFile.java

```
package tom.jiafei;
import java.io.*;
class FileExtendName implements FilenameFilter {
    String str=null;
    FileExtendName (String s) {
        str="."+s;
    }
    public boolean accept(File dir,String name) {
        return name.endsWith(str);
    }
}
```

6.3 bean的辅助类

ListFile.java

```
public class ListFile {  
    String extendsName=null;  
    String [] allFileName=null;  
    String dir=null;  
    public void setDir(String dir) {this.dir =dir;}  
    public String getDir() {return dir;}  
    public void setExtendsName(String s) {extendsName=s;}  
    public String getExtendsName() {return extendsName;}  
    public String [] getAllFileName() {  
        if(dir!=null) {  
            File mulu=new File(dir);  
            FileExtendName help=new FileExtendName(extendsName);  
            allFileName=mulu.list(help);  
        }  
        return allFileName;  
    }  
}
```

6.3 bean的辅助类

example4_6.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
<jsp:useBean id="file" class="tom.jiafei.ListFile" scope="request" />
<HTML><body bgcolor=cyan><font size=2>
<form action="" Method="post">
输入目录名, 例如:D:/2000<input type=text name="dir" size=15><br>
输入文件的扩展名,例如:java<input type=text name="extendsName" size=6>
  <input type=submit value="提交">
</form>
  <jsp:setProperty name="file" property="dir" param="dir" />
  <jsp:setProperty name="file" property="extendsName"
    param="extendsName" />
  <br><b>目录 <jsp:getProperty name="file" property="dir" />中
扩展名是 <jsp:getProperty name="file" property="extendsName" /> 的文件有:
  <% String [] fileName = file.getAllFileName();
    if(fileName!=null) {
      for(int i=0;i<fileName.length;i++) {
        out.print("<br>" + fileName[i]);
      }
    }
  %>
</font></body></HTML>
```

6.4 JSP和bean结合的简单例子

- JSP + JavaBean 模式开发，JSP页面中调用Bean将数据处理代码从页面中分离出来，实现代码复用，以便有效地维护一个Web应用。
- 结合一些具体的实际问题，熟悉掌握bean的使用。

6.4 JSP和bean结合的简单例子

Triangle.java

```
package red.star;
public class Triangle {
    double sideA=-1,sideB=-1,sideC=-1, area=-1;
    boolean triangle;
    public void setSideA(double a) {sideA=a;}
    public double getSideA() { return sideA;}
    public void setSideB(double b) {sideB=b;}
    public double getSideB() {return sideB;}
    public void setSideC(double c) {sideC=c;}
    public double getSideC() {return sideC;}
    public double getArea() {
        double p=(sideA+sideB+sideC)/2.0;
        if(triangle) area=Math.sqrt(p*(p-sideA)*(p-sideB)*(p-sideC));
        return area;}
    public boolean isTriangle()
    { if(sideA<sideB+sideC&&sideB<sideA+sideC&&sideC<sideA+sideB)
        triangle=true;
        else triangle=false;
        return triangle;
    }
}
```

6.4 JSP和bean结合的简单例子

example4_7.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
<jsp:useBean id="tri" class="red.star.Triangle" scope="request" />
<HTML><body bgcolor=#AAFF99><font size=3>
<form action="" method="post" >
  输入三角形三边:
  边A:<input type="text" name="sideA" value=0 size=5>
  边B:<input type="text" name="sideB" value=0 size=5>
  边C:<input type="text" name="sideC" value=0 size=5>
  <input type="submit" value="提交">
</form>
<jsp:setProperty name="tri" property="*" />
  三角形的三边是:
  <jsp:getProperty name="tri" property="sideA" />,
  <jsp:getProperty name="tri" property="sideB" />,
  <jsp:getProperty name="tri" property="sideC" />.
<br><b>这三个边能构成一个三角形吗? <jsp:getProperty name="tri"
  property="triangle" />
<br>面积是:<jsp:getProperty name="tri" property="area" /></b>
</font></body></HTML>
```

6.4 JSP和bean结合的简单例子

ComputerBean.java

```
package red.star;
public class ComputerBean {
    double numberOne,numberTwo,result;
    String operator="+";
    public void setNumberOne(double n) {numberOne=n;}
    public double getNumberOne() {return numberOne; }
    public void setNumberTwo(double n) {numberTwo=n;}
    public double getNumberTwo() {return numberTwo; }
    public void setOperator(String s) {operator=s.trim();}
    public String getOperator() {return operator;}
    public double getResult() {
        if(operator.equals("+"))
            result=numberOne+numberTwo;
        else if(operator.equals("-"))
            result=numberOne-numberTwo;
        else if(operator.equals("*"))
            result=numberOne*numberTwo;
        else if(operator.equals("/"))
            result=numberOne/numberTwo;
        return result; }
}
```

}

6.4 JSP和bean结合的简单例子

example4_8.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
<jsp:useBean id="computer" class="red.star.ComputerBean"
  scope="session"/>
<HTML><body bgcolor=yellow><font size=2>
  <jsp:setProperty name="computer" property="*" />
  <form action="" method=post name=form>
    <input type=text name="numberOne" size=6>
      <select name="operator" >
        <option value="+">+
        <option value="-">-
        <option value="*">*
        <option value="/">/
      </select>
    <input type=text name="numberTwo" size=6>
    <br> <input type="submit" value="提交" name="submit">
  </form>
  <b>
    <jsp:getProperty name="computer" property="numberOne" />
    <jsp:getProperty name="computer" property="operator" />
    <jsp:getProperty name="computer" property="numberTwo" /> =
    <jsp:getProperty name="computer" property="result" /> </b>
  </font></body></HTML>
```

6.4 JSP和bean结合的简单例子

Play.java

```
package red.star;
import java.io.*;
public class Play {
    int imageNumber=0,max;
    String pictureName[],playImage;
    String webDir="";        //web服务目录的名字
    String tomcatDir;        //tomcat的安装目录
    public Play() {
        File f= new File(""); //该文件认为在Tomcat引擎启动的目录中, 即bin目录中
        String path = f.getAbsolutePath();
        int index = path.indexOf("bin");
        tomcatDir = path.substring(0,index);
    }
    public void setWebDir(String s) {
        webDir =s;
        if(pictureName==null) {
            File dirImage=new File(tomcatDir+"/webapps/"+webDir+"/image");
            pictureName=dirImage.list();
        }
        if(pictureName!=null)
            max=pictureName.length;
    }
}
```

6.4 JSP和bean结合的简单例子

Play.java

```
public void setImageNumber(int n){
    if(n<0)
        n=max-1;
    if(n==max)
        n=0;
    imageNumber=n;
}
public int getImageNumber() {
    return imageNumber;
}
public String getPlayImage() {
    if(pictureName!=null)
        playImage=pictureName[imageNumber];
    return playImage;
}
}
```

6.4 JSP和bean结合的简单例子

example4_9.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
```

```
<jsp:useBean id="play" class="red.star.Play" scope="session" />
```

```
<%
```

```
String webDir = request.getContextPath();//获取当前Web服务目录  
的名称
```

```
webDir = webDir.substring(1); //去掉名称前面的目录符号: /
```

```
%>
```

```
<jsp:setProperty name="play" property="webDir" value="<%=  
webDir %>" />
```

```
<jsp:setProperty name="play" property="imageNumber"  
param="imageNumber"/>
```

6.4 JSP和bean结合的简单例子

example4_9.jsp

```
<HTML><body bgcolor=cyan><font size=2>
  <image src = image/ <jsp:getProperty name="play"
    property="playImage"/>
    width =120 height =90><br>这是当前显示的图像</image>
  <br>单击"上一张"或"下一张"按钮浏览图像
  <form action="" method=post>
    <input type=submit name="ok" value="上一张">
    <input type="hidden" name="imageNumber"
    value="<%=play.getImageNumber()-1%>">
  </form>
  <form action="" method=post>
    <input type=submit name="ok" value="下一张">
    <input type="hidden" name="imageNumber"
    value="<%=play.getImageNumber()+1 %>">
  </form>
</font></body></HTML>
```


6.4 JSP和bean结合的简单例子

ComputerCount.java

```
package red.star;
import java.io.*;
public class ComputerCount {
    int number=0;
    boolean isCome=false;
    File file=new File("count.dat") ;
    public int getNumber(){
        if(isCome==false)
            countPeople();
        return number;
    }
}
```

6.4 JSP和bean结合的简单例子

ComputerCount.java

```
private void countPeople() {  
    if(!file.exists()) {  
        number++;  
        try { file.createNewFile();  
            FileOutputStream out=new FileOutputStream(file);  
            DataOutputStream dataOut=new DataOutputStream(out);  
            dataOut.writeInt(number);  
            out.close();  
            dataOut.close();  
        }catch(IOException ee){}  
    }else {  
        try{ FileInputStream in=new FileInputStream(file);  
            DataInputStream dataIn=new DataInputStream(in);  
            number=dataIn.readInt();  
            number++;  
            in.close();  
            dataIn.close();  
            FileOutputStream out=new FileOutputStream(file);  
            DataOutputStream dataOut=new DataOutputStream(out);  
            dataOut.writeInt(number);  
            out.close();  
            dataOut.close();  
        }catch(IOException ee){}  
    }  
    isCome=true; } }
```

6.4 JSP和bean结合的简单例子

example4_10_a.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
```

```
<jsp:useBean id="count" class="red.star.ComputerCount"  
    scope="session"/>
```

```
<HTML><body bgcolor=cyan><font size=3>
```

```
<p> 网站的example4_10_a.jsp页面
```

```
<br>您是第<jsp:getProperty name="count" property="number"/>  
    个访问者。<br>
```

```
<a href="example4_10_b.jsp">欢迎去example4_10_b.jsp参观  
    </a>
```

```
</body></HTML>
```

6.4 JSP和bean结合的简单例子

example4_10_b.jsp

```
<%@ page contentType="text/html;charset=gb2312" %>
<jsp:useBean id="count" class="red.star.ComputerCount"
  scope="session"/>
<HTML><body><font size=3>
<p> 网站的example4_10_b.jsp页面
  <br>您是第<jsp:getProperty name="count" property="number"/>
  个访问者。<b>
  <a href="example4_10_a.jsp">欢迎去example4_10_a.jsp参观
  </a>
</body></HTML>
```

6.5 小结

- Javabeen是一个可重复使用的软件组件，是遵循一定标准、用Java语言编写的一个类，该类的一个实例称作一个Javabeen。
- 一个JSP页面可以将数据的处理过程指派给一个或几个bean来完成，我们只需在JSP页面中调用这个bean即可。在JSP页面中调用bean可以将数据的处理代码从页面中分离出来，实现代码服用，更有效维护一个Web应用。
- bean的生命周期分为page、request、session和application。