**JavaBean**

**一般可分为两个方向：数据Bean和业务逻辑Bean**

**数据bean：实现对表单输入的捕获、保存，减少对数据库的访问，获奖数据Bean放在一定作用域内，使此作用域内的多个JSP页面共享；**

**业务逻辑Bean：完成操作数据库、处理数据等业务逻辑，以数据bean或页面传递的值为参数**

MVC设计模式：（Model　View  Controller）  Class   jsp    Servlet

**JavaBean**是一种可重复使用，且跨平台的软件组件。JavaBean可分为两种：一种是有用户界面（UI）的javaBean；还有一种是没有用户界面，主要负责处理事务（如数据运算，操纵数据库）的javaBean。Jsp通常访问的是后一种javaBean.

**Jsp 和javaBean搭配使用的优点：**

1.使得html与java程序分离，这样便于维护代码。如果把所有的程序代码都写到jsp网页中，会使用代码繁杂，难以维护。

2.可以降低开发jsp网页人员对java编程能力的要求。

3.jsp侧重于生成动态网页，事务处理由javaBean来完成，这样可以充分利用javaBean组件的可重用性特点，提高开发网站的效率。

**一个标准的javaBean有以下几个特性:**

1.javaBean是一个公共的类

2.javaBean有一个不带参数的构造方法

3.javaBean通过setXXX方法设置属性，通过getXXX方法获取属性。

|  |
| --- |
| **package** com.anllin.bean;  **public** **class** Person  {  **private** String name;  **private** **int** age;  **private** String address;  **public** String getName() { **return** name; }  **public** **void** setName(String name) { **this**.name = name; }  **public** **int** getAge() { **return** age; }  **public** **void** setAge(**int** age) { **this**.age = age; }  **public** String getAddress() { **return** address;}  **public** **void** setAddress(String address){**this**.address = address;}  } |

**Jsp访问javaBean的语法**

1.导入javaBean类

|  |
| --- |
| <%@ page import=*"com.anllin.bean.Person"*%> |

2.声明javaBean对象

|  |
| --- |
| <jsp:useBean id=*"person"* class=*"com.anllin.bean.Person"*></jsp:useBean> |

其实相当于

|  |
| --- |
| <%          Person person = **new** Person();      %> |

Id表示生成一个类的实例，一般id不能重复，用于表示不同的对象，如果相同则表示同一个对象，这种情况下jsp会报错。

3.访问javaBean属性

|  |
| --- |
| <jsp:setProperty property=*"name"* name=*"person"* value=*"jack"*/>    <jsp:getProperty property=*"name"* name=*"person"*/><br> |

其实相当于

|  |
| --- |
| <%          person.setName("jack");          person.getName();      %> |

完整的例子：

|  |
| --- |
| <%@ page language=*"java"* import=*"com.anllin.bean.Person"*pageEncoding=*"UTF-8"*%>  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  <html>    <head>      <title>My JSP 'javabean.jsp' starting page</title>    </head>    <body>  <jsp:useBean id=*"person"* class=*"com.anllin.bean.Person"*></jsp:useBean>      <jsp:setProperty property=*"name"* name=*"person"* value=*"jack"*/>      <jsp:setProperty property=*"age"* name=*"person"* value=*"20"*/>    <jsp:setProperty property=*"address"* name=*"person"* value=*"beijing"*/>      <jsp:getProperty property=*"name"* name=*"person"*/><br>      <jsp:getProperty property=*"age"* name=*"person"*/><br>      <jsp:getProperty property=*"address"* name=*"person"*/><br>    </body>  </html> |

**设置bean的属性值**

1.可以在<jsp:setProperty>中通过param参数为bean的属性动态赋值。

|  |
| --- |
| <jsp:setProperty property=*"age"* name=*"person"* param=*"sage"*/><br> |

sage表示参数名,而不是属性

在浏览器地址栏中输入

http://localhost:8080/test/javabean.jsp?sage=30

则可以得到age属性的值30

**Javabean的存活范围**

Scope属性决定了javabean对象的存在的范围。可选值有：

1.page(默认值)

2.request

3.session

4.application

|  |
| --- |
| <jsp:useBean id=*"person"* class=*"com.anllin.bean.Person"* scope=*"page"*></jsp:useBean> |

**Javabean在page范围内时**

客户每次请求访问jsp页面时，都会创建一个javabean对象。Javabean对象的有效范围是客户请求访问的当前jsp网页。Javabean对象的以下两种情况下都会结束生命期：

1.客户请求访问的是当前jsp网页通过<forward>标记将请求转发到另一个文件

2.客户请求访问的是当前jsp页面执行完毕并向客户端发回响应。

**Javabean在request范围内**

客户每次请求访问jsp页面时，都会创建新的javabean对象。Javabean对象的有效范围为：

1.客户请求访问的当前jsp网页。

2.和当前网页共享一个客户请求的网页，即当前jsp网页中<%@include %>指令以及<forward>标记包含的其他jsp文件

3.当所有共享同一个客户请求的jsp页面执行完毕并向客户端发加响应时，javabean对象结束生命周期。

4.javabean对象作为属性保存在httpRequest对象中，属性名javabean的id,属性值为javabean对象，因此可以通过HttpRequest.getAttribute()方法取得javabean对象，例如：

Jvabean.jsp

|  |
| --- |
| <jsp:useBean id=*"person"* class=*"com.anllin.bean.Person"*scope=*"request"*></jsp:useBean>  <jsp:forward page=*"javabean2.jsp"*></jsp:forward> |

Javabean2.jsp

|  |
| --- |
| <%@ page import=*"com.anllin.bean.Person"*%>      <%      Person person = (Person)request.getAttribute("person");      out.print(person.getName());      %> |

**Javabean在session范围内**

1.Javabean对象被创建后，它存在于整个session的生命周期内，现个session中的jsp文件共享这个javabean对象。

2.javabean对象作为属性保存在HttpSession对象中，属性名为javabean的id,属性值为javabean对象。除了可以通过javabean的id直接引用javabean对象外，也可以通过HttpSession.getAttribute（）方法取得javabean对象，例如：

|  |
| --- |
| <jsp:useBean id=*"person"* class=*"com.anllin.bean.Person"*scope=*"session"*></jsp:useBean> |

|  |
| --- |
| Person person = (Person)session.getAttribute("person");      out.print(person.getName()); |

**Javabean在application范围内**

1.javabean对象被创建后，它存在于整个web应用的生命周期内，web应用中的所有jsp文件都能共享同一个javabean对象。

2.javabean对象作为属性保存在application对象中，属性名为javabean的id，属性值为javabean对象，除了可以通过javabean的id直接引用对象外，也可以通过javabean的application.getAttribute()方法取得javabean对象，例如：

|  |
| --- |
| <jsp:useBean id=*"person"* class=*"com.anllin.bean.Person"*scope=*"application"*></jsp:useBean> |

|  |
| --- |
| Person person = (Person)application.getAttribute("person");      out.print(person.getName()); |

假如有一个Person类,有一个属性name

为什么我们要提供一个getName()方法和setName()方法，而不是getCCNma之类的？

因为只有遵守这个规范，才能通过反射去调用这两个方法。