



# JSP程序设计教程

---

## 第5章 JavaBean技术



# 第5章 JavaBean技术

---

- 5.1 JavaBean概述 ✓
- 5.2 JavaBean中的属性 ✓
- 5.3 JavaBean的应用 ✓
- 5.4 JavaBean的应用实例 ✓



## 5.1 JavaBean概述

---

JSP较其他同类语言最强有力的方面就是能够使用JavaBean组件，JavaBean组件就是利用Java语言编写的组件，它好比一个封装好的容器，使用者并不知道其内部是如何构造的，但它却具有适应用户要求的功能，每个JavaBean都实现了一个特定的功能，通过合理地组织不同功能的JavaBean，可以快速生成一个全新的应用程序。如果将一个应用程序比做一间空房间，那么这些JavaBean就好比房间中的家具。



## 5.1 JavaBean概述

---

5.1.1 JavaBean技术介绍 ✓

5.1.2 JavaBean的种类 ✓

5.1.3 JavaBean规范 ✓





## 5.1.1 JavaBean技术介绍

使用JavaBean的最大优点就在于它可以提高代码的重用性，例如正在开发一个商品信息显示界面，由于商品信息存放在数据库指定表中，此时需要执行连接数据库、查询数据库、显示数据操作，如果将这些数据库操作代码都放入JSP页面中，代码复杂度可以想象，非编程人员根本无法接收这样的代码，这将为开发带来极大的不便。

编写一个成功的JavaBean，宗旨是“一次性编写，任何地方执行，任何地方重用”，这正迎合了当今软件开发的潮流，“简单复杂化”，将复杂需求分解成简单的功能模块，这些模块是相对独立的，可以继承、重用，这样为软件开发提供了一个简单、紧凑、优秀的解决方案。



## 5.1.1 JavaBean技术介绍

---

### 1. 一次性编写

一个成功的JavaBean组件重用时不需要重新编写，开发者只需要根据需求修改和升级代码即可。

### 2. 任何地方执行

一个成功的JavaBean组件可以在任何平台上运行，由于JavaBean是基于Java语言编写的，所以它可以轻易移植到各种运行平台上。

### 3. 任何地方重用

一个成功的JavaBean组件能够被在多种方案中使用，包括应用程序、其他组件、Web应用等。





## 5.1.2 JavaBean的种类

---

最初，JavaBean主要应用于可视化领域，现在JavaBean更多应用于非可视化领域，并且在服务器端表现出卓越的性能。

JavaBean按功能可分为可视化JavaBean和不可视化JavaBean两类。

### （1）可视化JavaBean

可视化JavaBean就是具有GUI图形用户界面的JavaBean；不可视JavaBean就是没有GUI图形用户界面的JavaBean，最终对用户是不可见的，它更多地是被应用到JSP中。



## 5.1.2 JavaBean的种类

---

### (2) 不可视JavaBean

不可视JavaBean又分为值JavaBean和工具JavaBean。值JavaBean严格遵循了JavaBean的命名规范，通常用来封装表单数据，作为信息的容器。

#### 【例5-1】 值JavaBean示例

工具JavaBean则可以不遵循JavaBean规范，通常用于封装业务逻辑，数据操作等，例如连接数据库，对数据库进行增、删、改、查和解决中文乱码等操作。工具JavaBean可以实现业务逻辑与页面显示的分离，提高了代码的可读性与易维护性。例如，下面的JavaBean就是一个工具JavaBean，它用来转换字符串中的“<”与“>”字符。





## 5.1.2 JavaBean的种类

---

【例5-2】 工具JavaBean示例





## 5.1.3 JavaBean规范

---

通常一个标准的JavaBean需遵循以下规范：

- （1）实现java.io.Serializable接口；
- （2）是一个公共类；
- （3）类中必须存在一个无参数的构造函数；
- （4）提供对应的setXxx()和getXxx()方法来存取类中的属性，方法中的“Xxx”为属性名称，属性的第一个字母应大写。若属性为布尔类型，则可使用isXxx()方法代替getXxx()方法。



## 5.1.3 JavaBean规范

实现`java.io.Serializable`接口的类实例化的对象被JVM（Java虚拟机）转化为一个字节序列，并且能够将这个字节序列完全恢复为原来的对象，序列化机制可以弥补网络传输中不同操作系统的差异问题。例如，当一台计算机在Windows系统上创建了一个对象，将这个对象序列化，并且通过网络将它发送到一台操作系统为Linux的计算机上，这时不必担心因为操作系统不同，传输的对象会有所改变，因为这个对象会重新准确组装。

作为JavaBean，对象的序列化也是必须的。使用一个JavaBean时，一般情况下是在设计阶段对它的状态信息进行配置，并在程序启动后期恢复，这种具体工作是由序列化完成的。



## 5.1.3 JavaBean规范

---

说明：如果在JSP中使用JavaBean组件，创建的JavaBean不必实现java.io.Serializable接口仍然可以运行。

### 【例5-3】 JavaBean规范示例





## 5.2 JavaBean中的属性

---

通常JavaBean中的属性分为以下4种：

- (1) 简单属性 (Simple)
- (2) 索引属性 (Indexed)
- (3) 绑定属性 (Bound)
- (4) 约束属性 (Constrained)

其中绑定属性和约束属性通常在JavaBean的图形编程中使用，所以在这里不进行介绍，下面来介绍JavaBean中的简单属性和索引属性。



## 5.2 JavaBean中的属性

### 1. 简单属性（Simple）

简单属性就是在JavaBean中对应了简单的setXxx()和getXxx()方法的变量，在创建JavaBean时，简单属性最为常用。

在JavaBean中，简单属性的getXxx()与setXxx()方法如下：

```
public void setXxx(type value);  
public type getXxx();
```

其中type表示属性的数据类型，若属性为布尔类型，则可使用isXXX()方法代替getXxx()方法。

【例5-4】 简单属性示例



## 5.2 JavaBean中的属性

### 2. 索引属性 (Indexed)

需要通过索引访问的属性通常称为索引属性。如存在一个大小为3的字符串数组，若要获取该字符串数组中指定位置中的元素，需要得知该元素的索引，则该字符串数组就被称为索引属性。

在JavaBean中，索引属性的getXxx()与setXxx()方法如下：

```
public void setXxx(type[] value);  
public type[] getXxx();  
public void setXxx(int index,type value);  
public type getXxx(int index);
```



## 5.2 JavaBean中的属性

---

其中**type**表示属性类型，第一个**setXxx()**方法为简单的**setXxx()**方法，用来为类型为数组的属性赋值，第二个**setXxx()**方法增加了一个表示索引的参数，用来为数组中索引为**index**的元素赋值为**value**指定的值；第一个**getXxx()**方法为简单**getXxx()**方法，用来返回一个数组，第二个**getXxx()**方法增加了一个表示索引的参数，用来返回数组中索引为**index**的元素值。

### 【例5-5】 索引属性示例







## 5.3 JavaBean的应用

---

5.3.1 创建JavaBean ✓

5.3.2 在JSP页面中应用JavaBean ✓





## 5.3.1 创建JavaBean

---

JavaBean实质上就是一种遵循了特殊规范的Java类，所以创建一个JavaBean，就是在遵循这些规范的基础上创建一个Java类。

在前面已经多次给出了JavaBean的代码，所以在这里不再给出代码进行讲解。首先新建一个记事本，然后输入代码，最后保存为\*.java文件即可完成一个JavaBean的创建。但通常都使用开发工具进行创建，如Eclipse。使用Eclipse开发工具创建JavaBean可以使用工具提供的功能自动生成属性的getXxx()与setXxx()方法，下面介绍如何在Eclipse中创建JavaBean。

【例5-6】 在Eclipse下创建JavaBean



## 5.3.2 在JSP页面中应用 JavaBean

在JSP中通常应用的是不可视JavaBean，它又分为值JavaBean和工具JavaBean。本节将来介绍如何在JSP页面中应用这两种JavaBean。

无论哪一种JavaBean，当它们被编译成Class文件后，需要放在项目中的WEB-INF\classes目录下，才可以在JSP页面中被调用。

### 1. 在JSP页面中应用值JavaBean

值JavaBean作为信息的容器，通常用来封装表单数据，也就是将用户向表单字段中输入的数据存储到JavaBean对应的属性中。使用值JavaBean可以减少在JSP页面中嵌入大量的Java代码。

## 5.3.2 在JSP页面中应用 JavaBean

### 【例5-7】 在JSP页面中应用值JavaBean

例如，存在一个登录页面，如图1所示。当用户输入用户名和密码进行登录后，要求在另一个页面中输出用户输入的用户名和密码，如图2所示。



图1

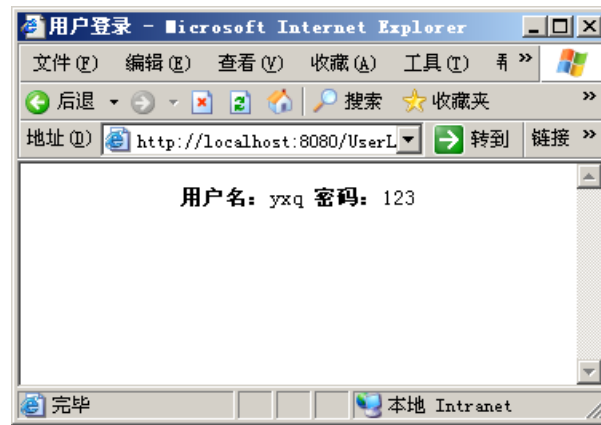


图2



## 5.3.2 在JSP页面中应用 JavaBean

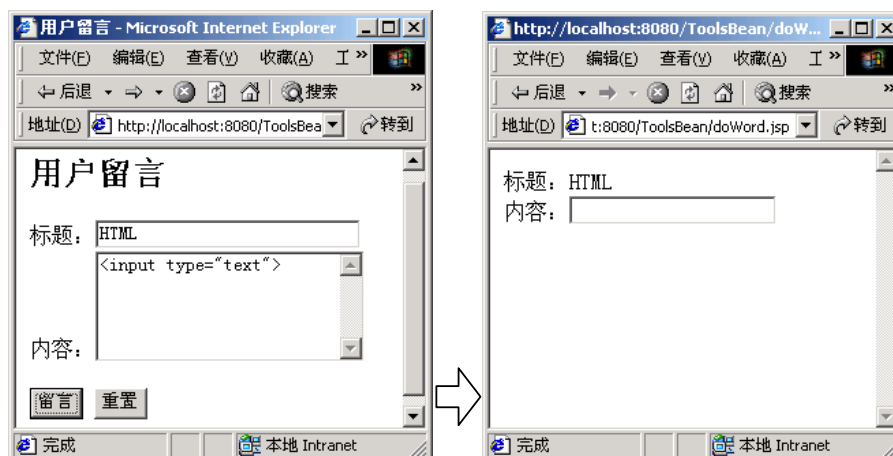
### 2. 在JSP页面中应用工具JavaBean

工具JavaBean通常用于封装业务逻辑、数据操作等，例如连接数据库，对数据库进行增、删、改、查和解决中文乱码等操作。使用工具JavaBean可以实现业务逻辑与前台程序的分离，提高了代码的可读性与易维护性。

#### 【例5-8】 在JSP页面中应用工具JavaBean

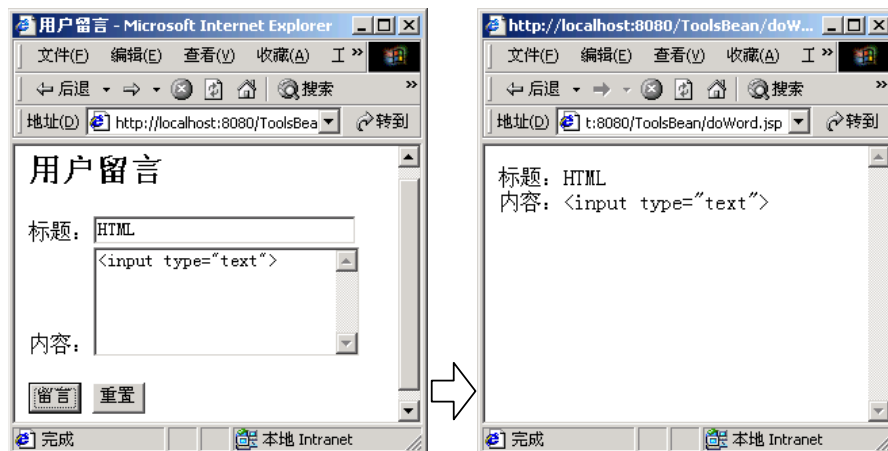
例如，在实现用户留言功能时，要将用户输入的留言标题和留言内容输出到页面中。若用户输入的信息中存在HTML语法中的“<”和“>”标识，如输入<input type= “text” >，则将该内容输出到页面后，会显示一个文本框，如下图所示；

## 5.3.2 在JSP页面中应用 JavaBean



但预先设想的是原封不动地输出用户输入的内容。解决该问题的方法是在输出内容之前，将内容中的“<”和“>”等HTML中的特殊字符进行转换，如将“<”转换为“&lt;”，将“>”转换为“&gt;”，这样当浏览器遇到“&lt;”时，就会输出“<”字符，如下图所示。

## 5.3.2 在JSP页面中应用 JavaBean





## 5.4 JavaBean的应用实例

---

5.4.1 应用JavaBean解决中文乱码 ✓

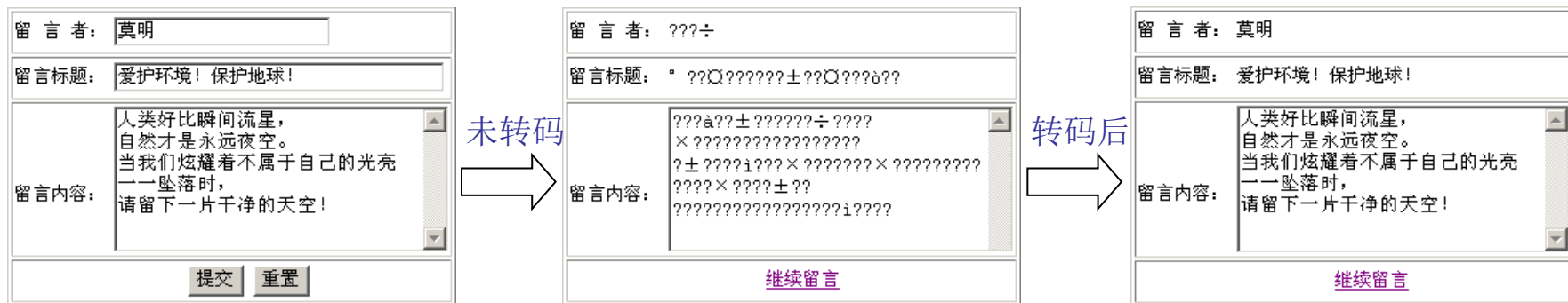
5.4.2 应用JavaBean实现购物车 ✓





## 5.4.1 应用JavaBean解决中文乱码

在JSP程序开发中，通过表单提交的数据中若存在中文，则获取该数据后输出到页面中将显示乱码，如下图所示。所以在输出获取的表单数据之前，必须进行转码操作。将该转码操作在JavaBean中实现，可在开发其他项目时重复使用，避免了重复编码。



## 5.4.1 应用JavaBean解决中文乱码

下面通过一个实例来介绍如何应用JavaBean解决中文乱码问题。

**【例5-9】** 应用JavaBean解决中文乱码



## 5.4.2 应用JavaBean实现购物车

购物车相信大家都已经非常熟悉，在现实生活中，购物车是商场提供给顾客用来存放自己所挑选的商品工具，顾客还可以从购物车中拿出不打算购买的商品。在Web程序开发中，购物车的概念被应用到了网络电子商城中，用户同样可对该购物车进行商品的添加和删除操作，并且购物车会自动计算出用户需要交付的费用。

本节将介绍应用JavaBean实现一个简单购物车的实例，该购物车实现了商品的添加、删除和清空所有商品的功能。

### 【例5-10】 应用JavaBean实现购物车

