



链滴

实例带你理解 JavaBean

作者: [adlered](#)

原文链接: <https://ld246.com/article/1552370060158>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



在我理解看来，JavaBean与[上文讲的接口](#)有着相似的地方：

- 用于包装并传递一些特定的数据（如字符串、整数、类等等）
- 用于实现方法的传递
-

在我们获取用户上传的数据时，如果只是用一个个变量去接收的话，就会造成重复代码的书写，并代码多且繁杂，如果想要在其他地方使用这个变量就会很不好调用；假设需要用户上传的数据较多，一次调用时都需要进行大量重复代码的书写，极大地影响了我们开发的效率，如果我们有一个可以去收这些数据的类或者容器就可以实现代码的“高内聚、低耦合”。

这个容器就被称为JavaBean，JavaBean里的属性全都使用private修饰可以保证只能通过JavaBean的getter方法和setter方法去设置或者获得类里的属性，当然setter和getter方法都是由public修饰，保其他地方能正常使用这些属性

JavaBean的设计规则：

JavaBean是一个公共类

JavaBean类具有一个公共无参的构造方法

JavaBean所有的属性定义为私有

JavaBean中，需要对每个属性都提供两个公共方法，set方法和get方法
定义JavaBean的时候，通常放置在一个命名的包下。

举个栗子

```
public class UserBean {
    //定义UserBean的私有对象 name和age
    private String name; //用户名
    private int age; //用户年龄

    //以下都是JavaBean的set方法和get方法
    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}
```

这是一个简单的JavaBean用来**存储用户数据**，它内置的属性很简单，只有**name**和**age**，也有关于属的**set**方法和**get**方法

```
public static void main(String[] args) {
    UseUserBean useBean = new UseUserBean(); //实例化UseUserBean对象，方便调用run方法
    UserBean user = new UserBean();
    user.setName("zhangsan");
    user.setAge(12);
    useBean.run(user);

    useBean.run2("zhangsan", 12);
}

//使用javabean的方式处理数据
public void run(UserBean user) {
    System.out.println("用户名: " + user.getName() + " 年龄: " + user.getAge());
}

//使用普通的变量的方式处理数据
public void run2(String name, int age){
    System.out.println("用户名: " + name + " 年龄: " + age);
}
```

run和**run2**的结果是一样的，都可以打印出用户的信息，但是当用户的其他数据比较多时，是不是方里的参数就变得很多了，并且外部也不好调用。

如果我将用户的数据都封装到一个类里，再通过变量的传递，就可以将这个类里的其他数据流转到其要使用这些属性的地方，将用户的数据封装到一个容器里，这种操作就降低了代码的耦合性。

使用javabean有很多好处，比如

1. 提高代码的可复用性：对于通用的业务处理逻辑，数据库操作都可以封装在JavaBean里，通过调用JavaBean的属性和方法可快速进行程序设计
2. 程序易于开发维护：实现逻辑的封装，使事物处理和显示互不干扰
3. 支持分布式运用：多用JavaBean，尽量减少Java代码和HTML的混编
4. 可以便捷地传递数据

MVC里有Model模型的概念，就是将数据封装到模型里，这和JavaBean的概念是一样的，就是构建一个模型，然后照着样子将数据放到一个模型里，这就是实例化了一个对象。

总结

关于JavaBean的知识还有很多，这里只是简单地介绍JavaBean，我们会经常在JavaWeb开发中用到。