JavaBean

# JavaBean简介

## JavaBean是一种特殊的Java类。

* **JavaBean是一种特殊的 Java 类，它遵从一定的设计模式**，开发工具和其他组件可以根据这种模式来调用JavaBean。
* JavaBean可以设计得像Swing组件一样复杂，提供方法、属性、事件、以及供开发工具将它实例化到一个GUI容器中并对其属性和行为进行定制的特性。
* JavaBean也可以设计得很简单，仅仅提供一些方法和属性供外部环境和其他组件调用。
* **用作JavaBean的类必须具有一个公共的、无参数的构造方法**，这个方法可以是通过编译器自动产生的那个缺省构造方法。
* **JavaBean的属性通过遵循某种规范的公共方法暴露给外部**，外部的其他程序可以通过Java 的反射API 来查找JavaBean中遵循这一规范的方法，从而发现JavaBean中的属性。

# EJB：企业JavaBean

# JavaBean属性的介绍

## 属性修改器：setter方法如setName方法

## 属性访问器：getter方法getName方法

* JavaBean的属性与**普通Java类的属性**的概念不一样，**JavaBean的属性是以方法定义的形式出现的**。
* 用于对属性赋值的方法称为**属性修改器或setter方法**，用于读取属性值的方法称为**属性访问器或getter方法**。
* 属性修改器必须以小写的set 前缀开始，**后跟属性名**，**且属性名的第一个字母要改为大写**，例如，nickName属性的修改器名称为setNickName，password属性的修改器名称为setPassword。
* **属性访问器**通常以小写的get 前缀开始，后跟属性名，且属性名的第一个字母要改为大写，例如，nickName属性的访问器名称为getNickName，password属性的访问器名称为getPassword。
* **JavaBean的属性名是根据setter方法与getter方法的名称来生成的**， setter方法或getter方法中除去前缀“set”和“get”后的部分即为属性名，但**属性名的首字母必须小写**。
* 属性分类：**只读**/**读写/只写**，简单/复杂

# JavaBean示例;

public class RegisterBean

{

**private String nickName = null;//属性**

**private boolean male = false;//属性**

public void setNickName(String nickName)

{

this.nickName = nickName;

}

public String getNickName()

{

return nickName;

}

public void setMale(boolean male)

{

this.male = male;

}

public boolean isMale()

{

return male;

}

}

# 带有索引的JavaBean（使用的不多）



