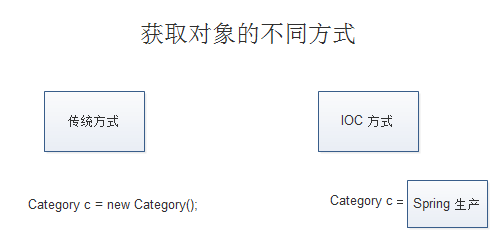
Spring的思想：

以获取对象的方式来进行比较  
  
****传统的方式：****  
通过new 关键字主动创建一个对象  
****IOC方式****  
对象的生命周期由Spring来管理，直接从Spring那里去获取一个对象。 IOC是****反转控制**** (Inversion Of Control)的缩写，就像控制权从本来在自己手里，交给了Spring。   
  
打个比喻：  
****传统方式：****相当于你自己去菜市场new 了一只鸡，不过是生鸡，要自己拔毛，去内脏，再上花椒，酱油，烤制，经过各种工序之后，才可以食用。  
****用 IOC：****相当于去馆子(Spring)点了一只鸡，交到你手上的时候，已经五味俱全，你就只管吃就行了。



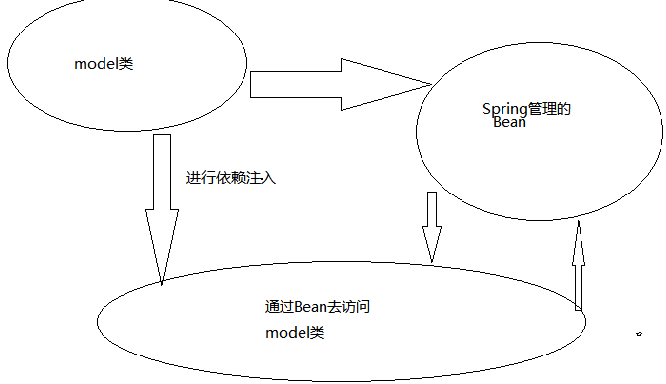
总结：spring中的bean有点像javabean 但是作用域不同。

在进行bean的书写时，要区分大小写，<bean name=”” class=””>

<property name=”id” value=””>

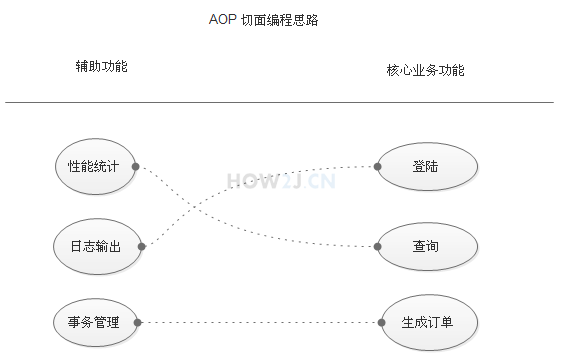
<bean/>

详细去看https://www.cnblogs.com/shinubi/p/4182027.html



AOP 即 Aspect Oriented Program 面向切面编程   
首先，在面向切面编程的思想里面，把功能分为****核心业务功能****，和****周边功能****。   
所谓的核心业务，比如登陆，增加数据，删除数据都叫核心业务   
所谓的周边功能，比如性能统计，日志，事务管理等等   
周边功能在Spring的面向切面编程AOP思想里，即被定义为****切面****   
在面向切面编程AOP的思想里面，核心业务功能和切面功能分别****独立进行开发****   
然后把切面功能和核心业务功能 ****"编织"**** 在一起，这就叫AOP

1. 功能分两大类，辅助功能和核心业务功能  
2. 辅助功能和核心业务功能****彼此独立****进行开发  
3. 比如登陆功能，即便是没有性能统计和日志输出，也可以正常运行  
4. 如果有需要，就把"日志输出" 功能和 "登陆" 功能 ****编织****在一起，这样登陆的时候，就可以看到日志输出了  
5. 辅助功能，又叫做****切面****，这种能够****选择性的，低耦合的****把切面和核心业务功能结合在一起的编程思想，就叫做切面编程



@Aspect 注解表示这是一个切面  
@Component 表示这是一个bean,由Spring进行管理  
@Around(value = "execution(\* com.how2java.service.ProductService.\*(..))") 表示对com.how2java.service.ProductService 这个类中的所有方法进行切面操作