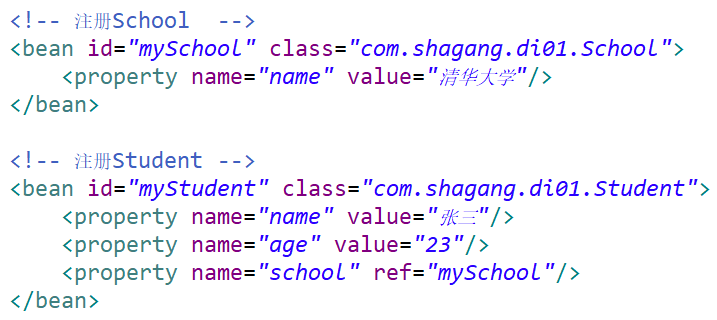
###### 注入分类

Bean实例在调用无参构造器创建了空值对象后，就要对Bean对象的属性进行初始化。初始化时由容器自动完成的，称为注入。根据注入方式的不同，常用的有两类：设值注入、构造注入。

还有一种，实现特定接口注入。由于这种方式采用了侵入式编程，污染了代码，所以几乎不用。

1. set方法注入（常用）

设值注入是指。通过setter方法传入被调用者的实例。这种注入方式简单、直观，因而在Spring的依赖注入中大量使用。



为mySchool对象中名为“name”的属性注入“清华大学”作为值；

为myStudent对象中名为“name”的属性注入“张三”，

“age”的属性注入“23”，

“school”的属性注入“mySchool”。

1. 构造注入

###### 命名空间注入（了解）

1. p命名空间设值注入
2. c命名空间构造注入

###### 集合属性注入（重点）



###### 对于域属性的自动注入

对于域属性的注入，也可不在配置文件中显示的注入。可以通过为<bean/>标签设置autowire属性值，为域属性进行隐式自动注入。根据自动注入判断标准的不同，可以分为两种：

1. byName：根据名称自动注入
2. byTyppe：根据类型自动注入

###### 使用SPEL注入

SPEL，Spring Expression Language，即Spring EL表达式。即，在Spring配置文件中为Bean的属性注入值时，可直接使用SPEL表达式计算的结果。SPEL表达式以#开头，后跟一对大括号。用法：<bean id=”abc” value=”#{......}”/>。

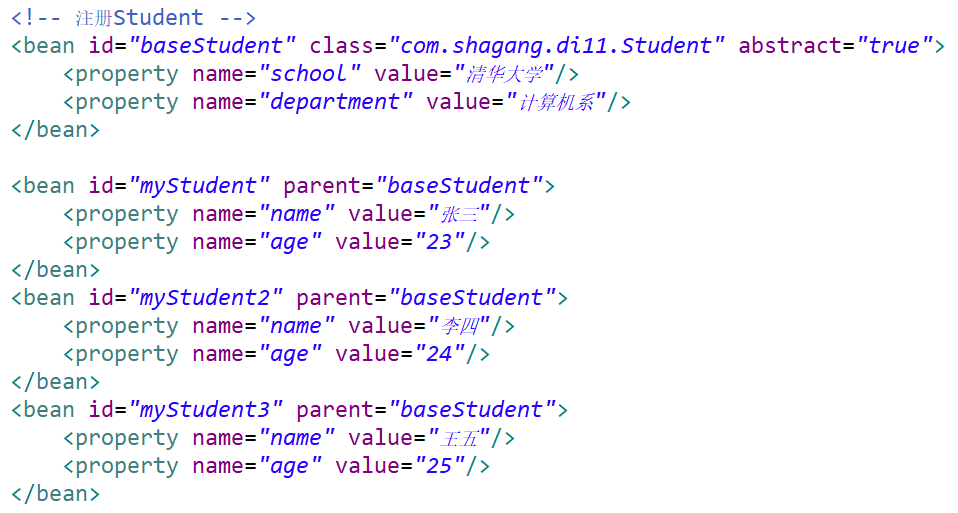
###### 使用内部Bean注入

若不希望代码直接访问某个bean，即，在代码中通过getBean方法获取该Bean实例，则可将该Bean的定义放入调用者bean定义的内部。

###### 使用同类抽象Bean注入

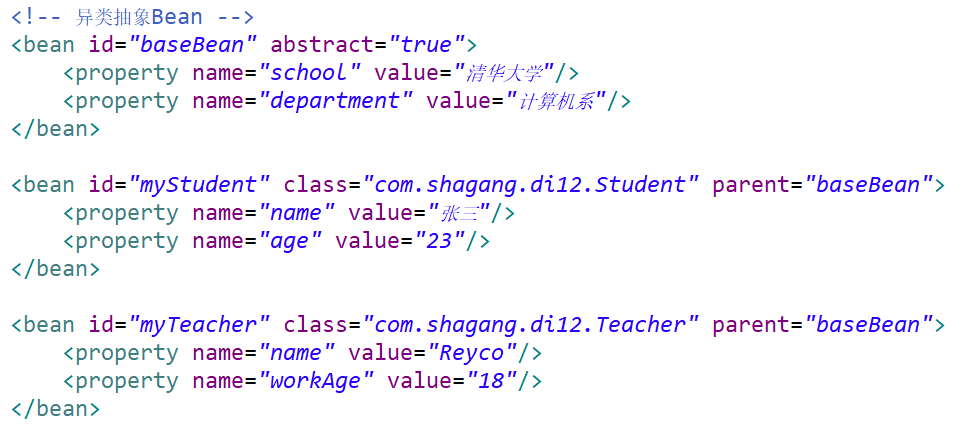
当若干Bean实例同属于一个类，且这些实例的属性值又有相同值时，可以使用抽象Bean，以简化配置文件。

抽象Bean是用于让其他bean继承的。这个bean在Bean类中是不能通过getBean方法获取的。设置abstract属性为true来指明该bean为抽象bean，默认值为false。不过，该bean不为抽象bean时，也可以被继承。只不过，在应用中，用于被继承的bean一般为抽象bean。



###### 使用异类抽象Bean注入

当若干不同的类对象具有相同的属性，且其值也相同时，可使用异类抽象Bean。



###### 为应用指定多个Spring配置文件（重点）

在实际应用里，随着应用规模的增加，系统中Bean数量也大量增加，导致配置文件变得非常庞大、臃肿。为了避免这种情况的产生，提高配置文件的可读性与可维护性，可以将Spring配置文件分解成多个配置文件。

1. 平等关系的配置文件

将配置文件分解为地位平等的多个配置文件，并将所有配置文件的路径定义为一个Spring数组，将其作为容器初始化参数出现。其将与可变参数的容器构造器匹配。



1. 包含关系的配置文件

