## Spring IOC底层原理

IOC的基本概念和原理：

基本概念：

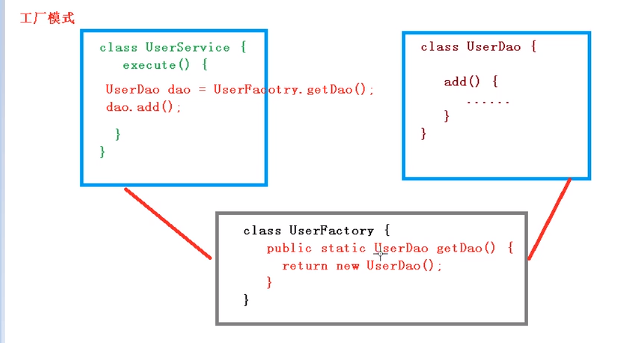
（1）控制反转，把对象的创建和对象之间的调用过程，都交给Spring管理；

（2）使用IOC的目的：为了降低耦合；

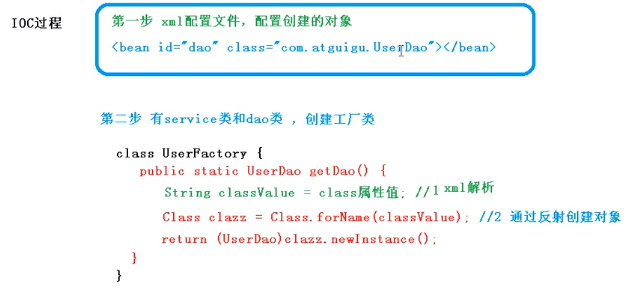
底层原理：主要用到了xml解析、工厂4模式、反射

工厂模式：假设UserService需要一个UserDao，就可以通过Factory来创建。假设不通过Factory，就需要在UserService中创建UserDao的对象。

目的是最大限度地降低耦合。



工厂模式和XML解析加上反射来得到对象：



进一步降低了耦合程度，此时，只需要修改xml文件就可以了。

IOC本质上就是一个容器，即对象工厂

IOC（接口）

1.IOC思想基于IOC容器完成，IOC容器底层就是对象工厂

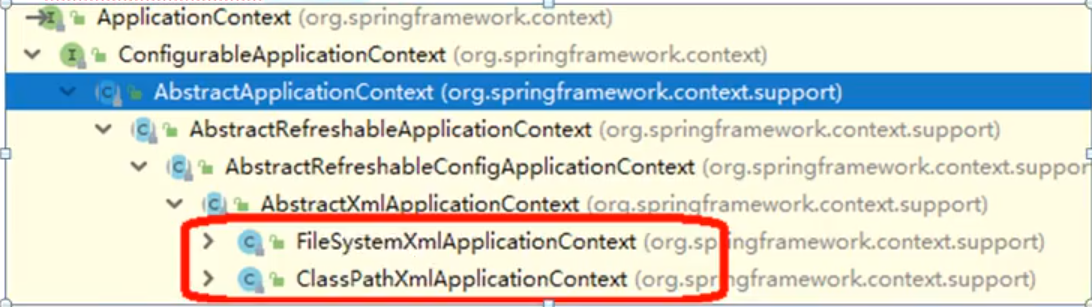
2.Spring提供IOC容器实现的两种方式：（两个接口）

（1）BeanFactory：IOC容器最基本的实现方式。是Spring内部使用的接口，一般不提供给开发人员使用；

（2）ApplicationContext：BeanFactory的子接口，提供的功能更加强大，一般由开发人员进行使用

（3）不同之处：BeanFactory 在加载配置文件的时候不会创建对象，在使用到对象的时候才会使用反射去创建；ApplicationContext在加载配置文件就会加载对象。

3.ApplicationContext的实现类：



上面的对应盘里面的路径，外面的对应src下的路径。

Bean管理操作的两种方式：

（1）基于xml配置文件方式

（2）基于注解的方式

注解：（1）注解是代码的特殊标记，格式@注解名称（属性名=属性值）；（2）注解可以在类上面，方法上面，属性上面；（3）使用注解的原因，简化xml配置，使用更简洁的方式配置xml。

Spring针对Bean管理创建对象提供的注解：

1）@Component

2）@Service

3）@Controller

4）@Repository

（各类注解一般用于各个层，但是只是建议，可以混用。功能都是一样的，都可以用来创建bean实例）

基于注解方式创建实例对象