[设计模式之—：控制反转IoC - 简书 (jianshu.com)](https://www.jianshu.com/p/5aaaf24e6c09)

[Spring系列之beanFactory与ApplicationContext - 平凡希 - 博客园 (cnblogs.com)](https://www.cnblogs.com/xiaoxi/p/5846416.html)（beanfactory）

**控制反转**（Inversion of Control，IoC）：调用者不再创建被调用者的实例，由容器创建。这很好理解，就是依赖的双方都不用负责类的实例化操作，我需要哪个对象，容器直接递上来，相当于多了一个管家，所谓衣来伸手、饭来张口，实例化这样的脏活累活全让容器包了。

**依赖注入**（Dependency Injection，DI）：容器创建好实例后再注入调用者称为依赖注入。简单说来，就是容器new完对象然后传递给调用方。Spring支持的注入方式主要有两种：setter注入(setter injection，调用者通过set方法设置被调用类)和构造器注入(constructor injection，调用者通过构造器设置被调用类)。

[(1条消息) Spring系列--spring IOC用到的设计模式\_张姐姐聊编程-CSDN博客](https://blog.csdn.net/zmh458/article/details/99685031)！！！！！！！

1.beanfactory实例化容器时并不会实例化对象，仅当调用指定bean时才会创建，创建方式如下：

BeanFactory beanFactory=new XmlBeanFactory(new ClassPathResource("com/service/stu/beans.xml"));//相对路径

ChangeLetter changeLetter1=(ChangeLetter) beanFactory.getBean(“changeLetter”);

2.applicationcontext实例化容器时，就会将配置在容器中的bean实例化，会消耗部分内存，但调用bean的过程会大大加快，创建过程如下：

ApplicationContext ac=new ClassPathXmlApplicationContext(“com/service/stu/beans.xml”);

实例化ac的另外两种方式：

FileSystemXmlApplicationContext，这种方式很鸡肋，需要使用绝对路径。

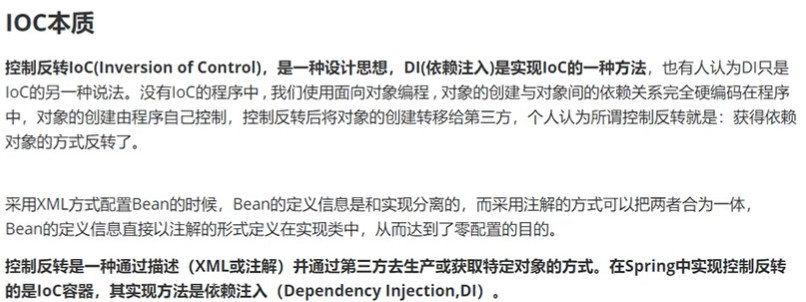
WebXmlApplicationContext，在web应用中主要通过这种方式来实例化bean容器，当中间件启动时该容器实例化。

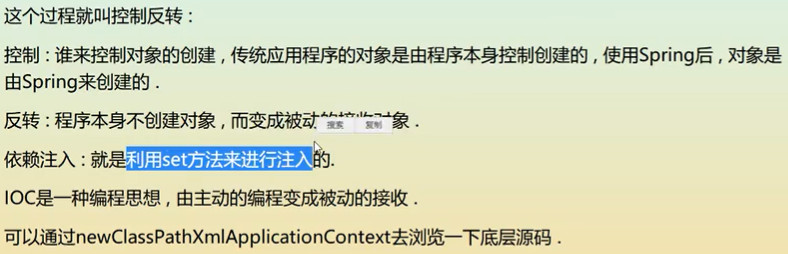
1、IOC基本思想发展：



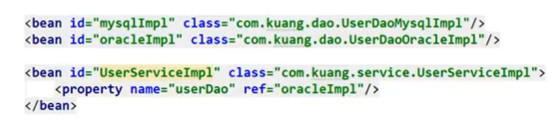
（所谓的解除了耦合性：就是我们不会再去依赖于具体的类而是在依赖于抽象类或者是接口，用户可以根据自己的需要进行选择，对象的创建当然就不再硬编码，不在程序员手中，而是**放到用户**中，所以将 程序更能应付用户需求的变更。）（IOC控制反转估摸就是这个意思）（但是用户能够自己去创建对象吗，当然不能，它只是用户，既然双方都不能创建，那就让第三方来进行代理，将对象的创建权交到第三方的手中。）

2、IOC本质：（首先控制反转是一**个抽象的概念或者理念，**而DI-依赖注入仅仅实现这种思想的一种方案。想想其实解耦的思想基本的就是中间添加额外的依赖的，不直接依赖而是采用间接的依赖，从而弱化两者之间的依赖关系）(对象由spring来创建、管理、装配)





3、示例：（其实spring与mybatis的自动依赖注入也是通过这种配置的一次注入进行实现。）





4、IOC的创建对象方式：

1、使用无参构造创建对象，默认。

2、假设我们要使用有参构造创建对象，则需要使用xml的bean进行构造器的配置。

1、下标赋值：

2、类型赋值：（基本类型可以使用类型，引用类型需要使用全限定名）--不建议使用

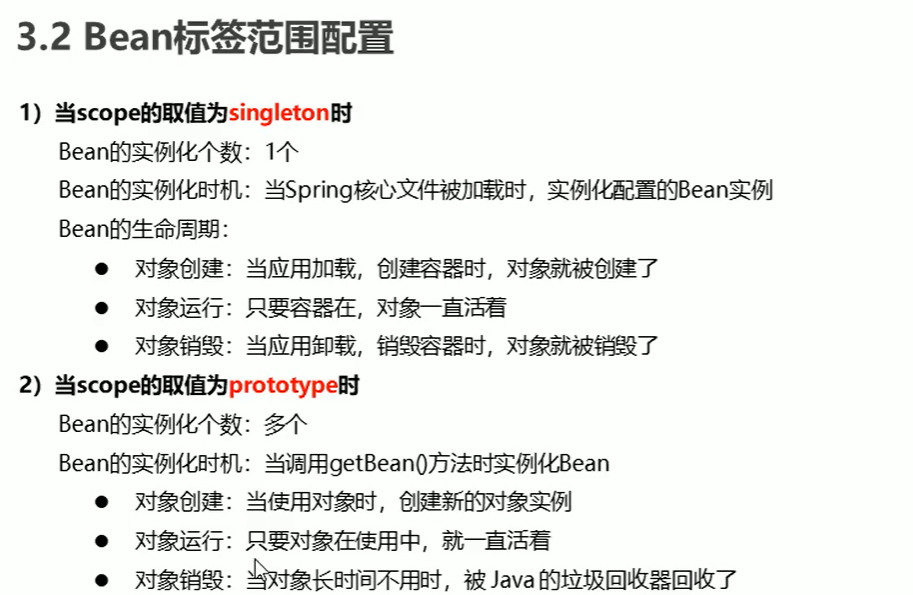
3、使用属性名。





总结：在配置文件加载的时候，容器中管理的对象就已经初始化了。

（如果bean是singleton则在加载配置文件时就已经被创建了，如果是prototype则是在获取时进行创建对象一次获取一次创建一次，创建对象默认使用无参构造器为啥不拷贝）



Spring配置：

1、别名



2、bean的配置：id在spring容器中的唯一标识符，class映射类的全限定名，name就是在spring容器中该bean的名字，可以取多个（使用分隔符进行分割，分隔符可以是很多种：空格，逗号，分号。。。。。。）。



（scope：标识该bean存活的时间。关于Bean的作用域(scope)。Spring容器最初提供了两种bean的scope类型：singleton和prototype,在Spring2.0之后又引入了另外三种scope类型：request、session、global session类型。不过这三种类型有所限制：只能在Web应用中使用。也就是说，只有Web应用的ApplicationContext中使用这三个scope才合理，类似如下代码来进行使用

[(1条消息) Spring Bean的scope(作用域)\_浅然的专栏-CSDN博客\_bean scope](https://blog.csdn.net/w_linux/article/details/80069039)

[单例和多例的区别 - 张亮java - 博客园 (cnblogs.com)](https://www.cnblogs.com/zhangliang88/p/5388472.html)）

3、导入import（一般用于团队开发使用，它可以将多个配置文件导入合并为一个）

(就是说我们的spring配置文件可以有很多个，每个配置文件可以配置一个或者多个bean，这样也方便分工与管理，最后我们可以通过一个bean配置文件，我们一般取名叫做applicationContext.xml来将所有的bean配置文件整合在一起，最后我们只需要读取一个配置文件获取到一个spring容器即可获取所有配置好的bean.)

