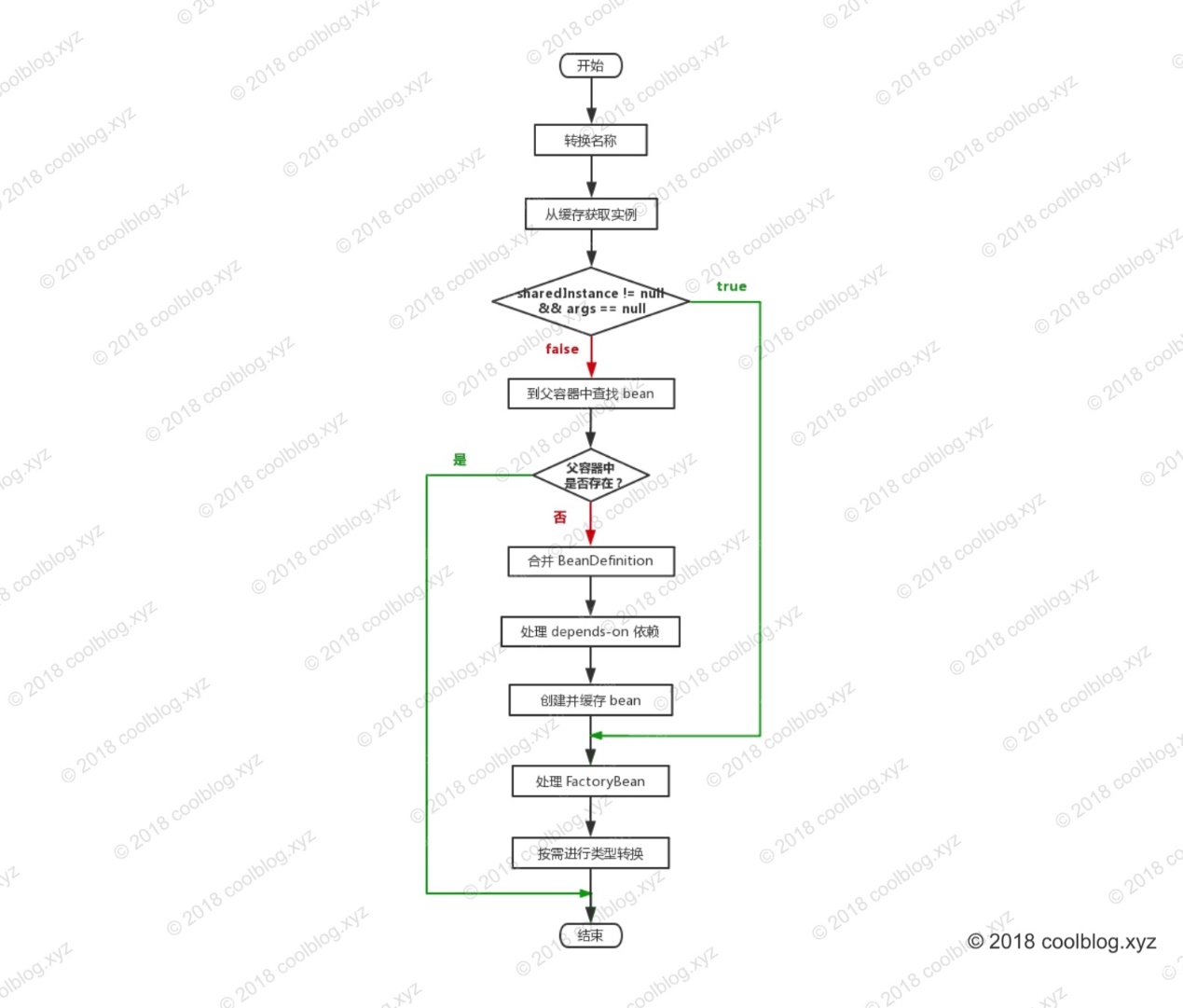
# SpringIOC 源码

方法入口：BeanFactory的getBean(String)方法 （Spring中加载bean为懒加载）

getBean调用doGetBean，doGetBean首先从缓存中获取单例bean，没有获取到，创建实例，在创建实例之前需要先去父容器里检查一下 ，然后检测是否有depensOn依赖，如果有则先初始化所依赖的bean，如果检测到有循环依赖，就抛异常。调用getSingleton方法创建bean实例，如果需要类型转换，接下来进行类型专管。



BeanName转换：对参数name进行转换，主要是为了解决两个问题，循环处理&字符，处理别名映射，对别名也要进行转换

从缓存中获取bean实例：对于单例bean Spring容器只会实例化一次，如果缓存中没有获取到，并且允许提前曝光，则从工厂类中获取一个初始bean（还未填充属性）并曝光，用于解决循环依赖

合并父BeanDefinition与子BeanDefinition：Spring支持配置继承，在parent属性配置父类bean（如果有父类配置，则递归合并）

从Factory中获取bean实例：判断bean是普通的bean还是FactoryBean类型的bean，普通Bean直接返回，FactoryBean类型先找缓存，再生成放缓存，然后进行后置处理。

# 创建Bean的流程

先从换从集合中获取bean实例，不为空则直接返回，若为空，进入创建bean实例阶段（向当前正在处于创建集合中注册）。调用匿名工厂类的getObject方法创建bean实例，将键值对缓存。

创建过程：解析bean类型，处理lookup-method和replace-method配置，在bean初始化前应用前后置处理。 从缓存中获取BeanWrapper实现类对象，并清理相关记录，若未命中则构建bean实例，将并将实例包裹在BeanWrapper实现类中，根据条件决定是否暴露早起引用，调用populateBean方法向bean实例中填充属性。

# 创建原始Bean对象

工厂方法创建对象；构造方法自动创建；默认构造方法创建

# 循环依赖的解决方法

暴露早期引用到早期引用集合，用于解决循环依赖问题

# 属性填充

获取属性列表

在属性被填充到bean前应用后置处理自定义填充 return

根据名称或属性注入依赖

后置处理

将属性应用到bean对象中（对String进行处理）

# 余下的初始化工作

根据实现的接口进行注入，调用前后置处理方法（AOP）