1. Bean的生命周期
2. 配置

通过配置文件定义Bean的信息，包括：类名、属性名、属性值、方法名等。

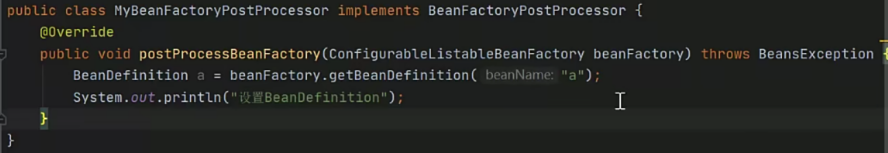
1. 解析

将配置文件内容通过对应的解析类（必须实现BeanDefinitionReader接口）解析为BeanDefinition对象，之后才能创建Bean对象。

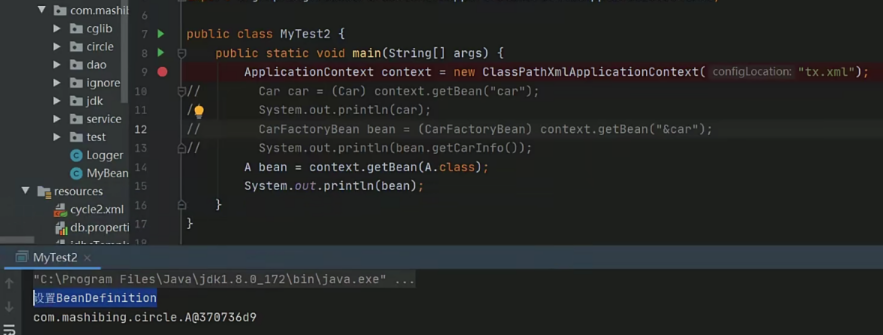
1. 增强BeanDefinition信息

为了扩展功能，容器创建过程动态改变BeanDefinition，可以进行属性值替换功能（填充占位符等等）。

自定义增强器：必须实现BeanFactoryPostProcessor接口



20210723230930



1. BeanFactory Bean工厂

通过BeanFactory工厂创建对象

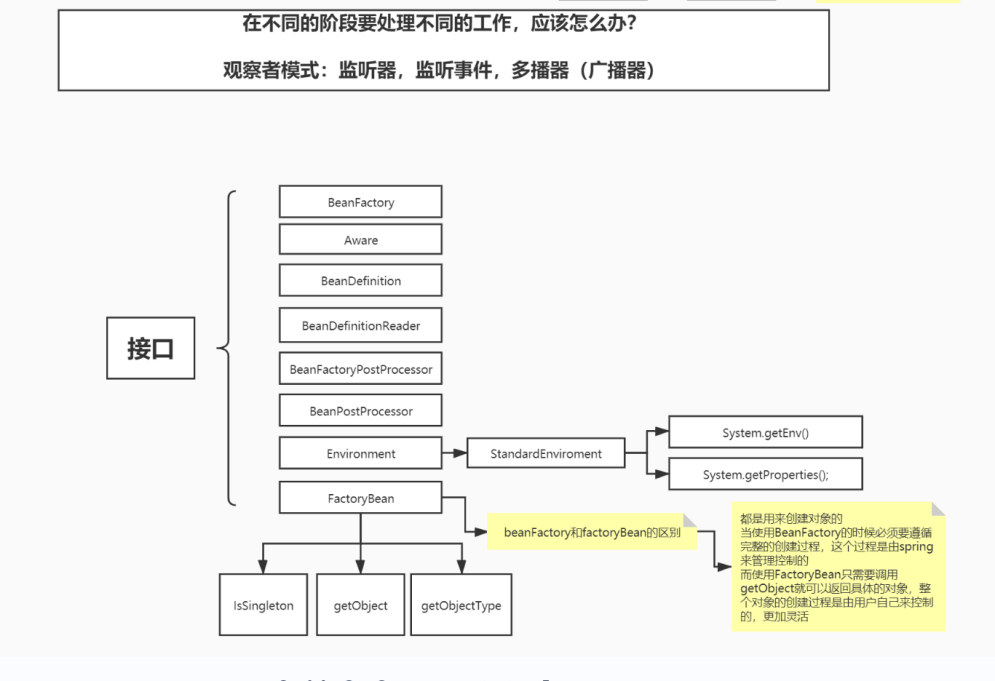
1. 创建对象
2. 实例化

在堆中为对象分配空间，并设置属性值为默认值。

1. 初始化

* 填充属性
* 设置Aware接口的属性
* Bean的前置增强器：BeanPostProcessorbefore，用于AOP（添加类属性方法等等）
* 执行Bean的init方法
* Bean的后置增强器：BeanPostProcessorafter，用于AOP（添加类属性方法等等）

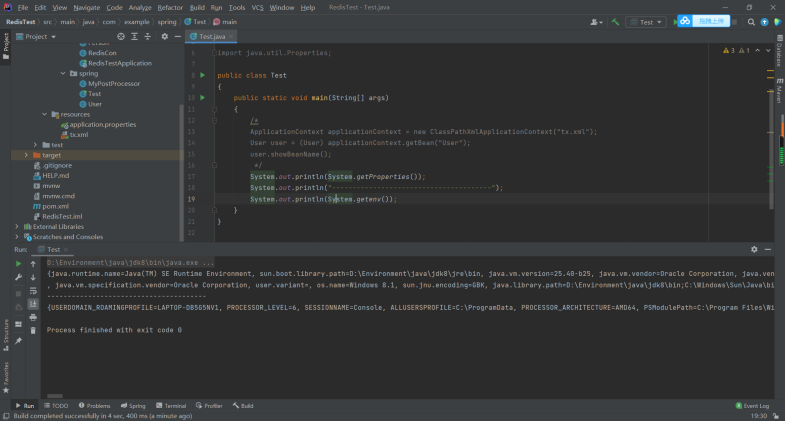
1. 常见接口



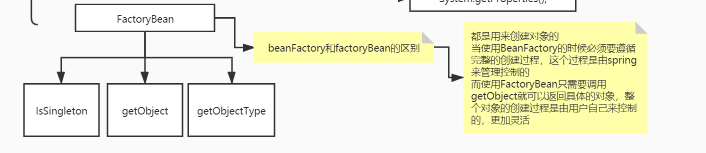
1. Environment

系统的属性值、变量值。

在容器框架还没启动的时候，把它添加到前置对象中，方便后续操作时直接从对象中获取，提高效率。



1. FactoryBean



相同点：BeanFactory和FactoryBean都可以创建对象。

不同点：

BeanFactory创建对象必须遵循完整的创建过程，该创建过程由spring管理控制；

（ctx.getBean(“beanName”)）

FactoryBean创建对象不需要遵循完整的创建过程，通过getObject()方法创建对象，该创建过程由用户自己控制。第一次获取需要创建，第二次获取直接从缓存中获取。（ctx.getBean(“factoryBeanName”)）【容器创建过程会创建FactoryBean，当调用factoryBean.getObject()方法才会创建bean对象】

