# Ioc

1.加载系统/环境变量Environment

2.创建BeanFactory,

加载解析xml propertis yml 解析器接口BeanDefinitionReader

把配置转为BeanDefiniton放入BeanDefinitionMap

如开启注解扫描,创建BeanFactory后置处理器 BeanDefiniton

创建Bean后置处理器 BeanDefiniton

3.执行BeanFactory后置处理器

1) ConfigurationClassPostProcessor

@Configuration @Bean @Component @ComponentScan @ImportResource

@Order @PropertySource @Import(自动装配处理) 生成配置BeanDefinition

bean实例化 开辟空间,属性值为默认值

bean初始化 1.填充属性

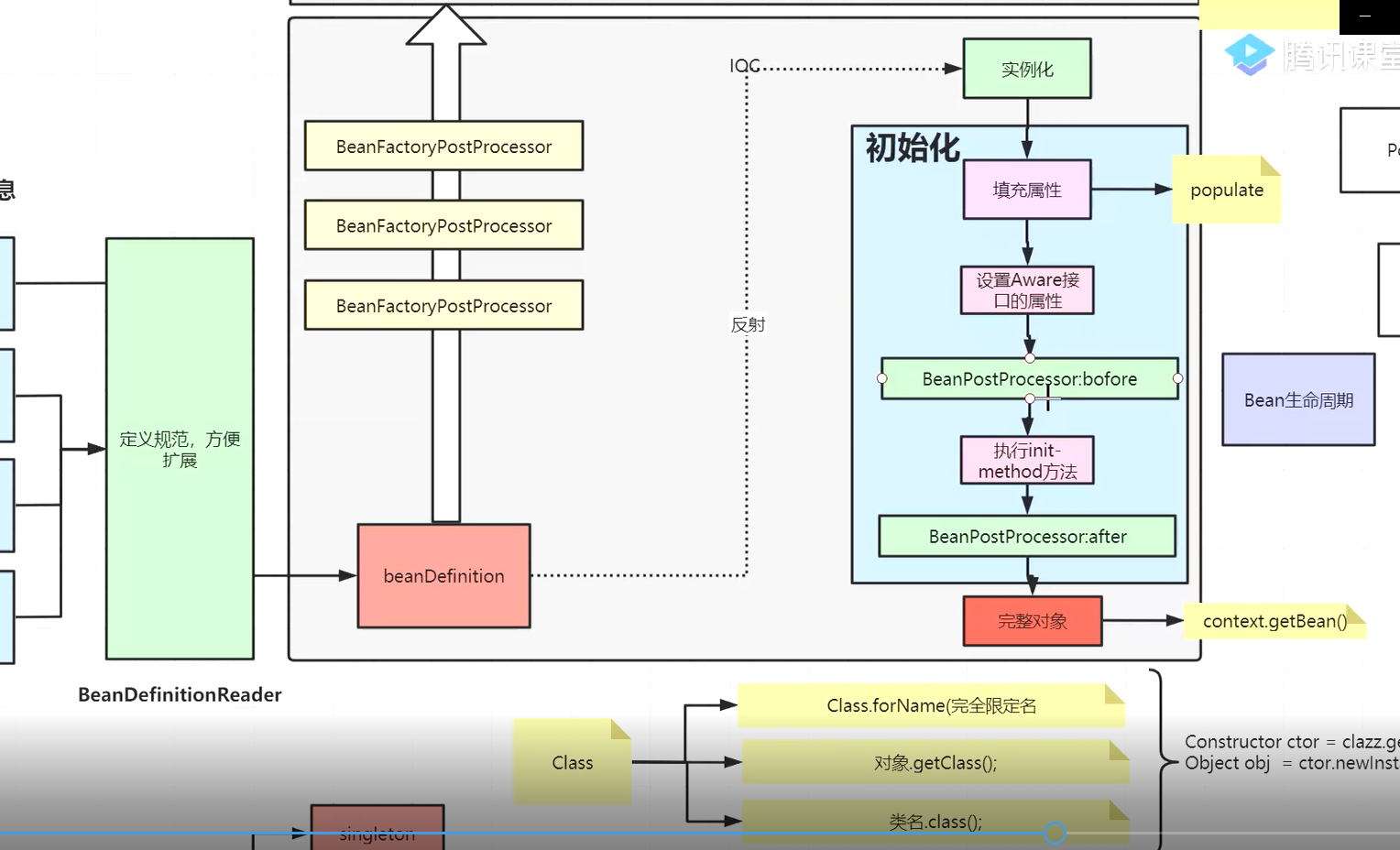
2.设置Aware接口 作用:为bean对象属性设置容器中对象

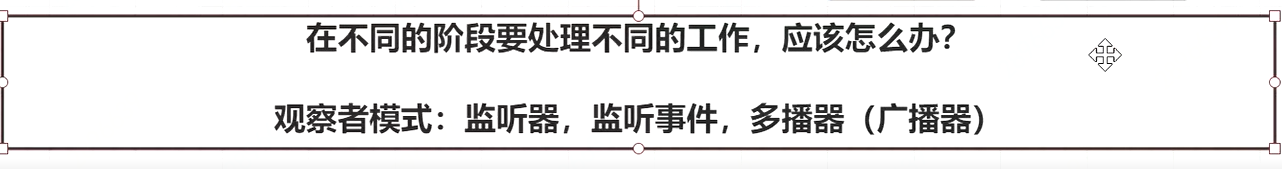
BeanPostProcessor:before 调用aware接口设置属性

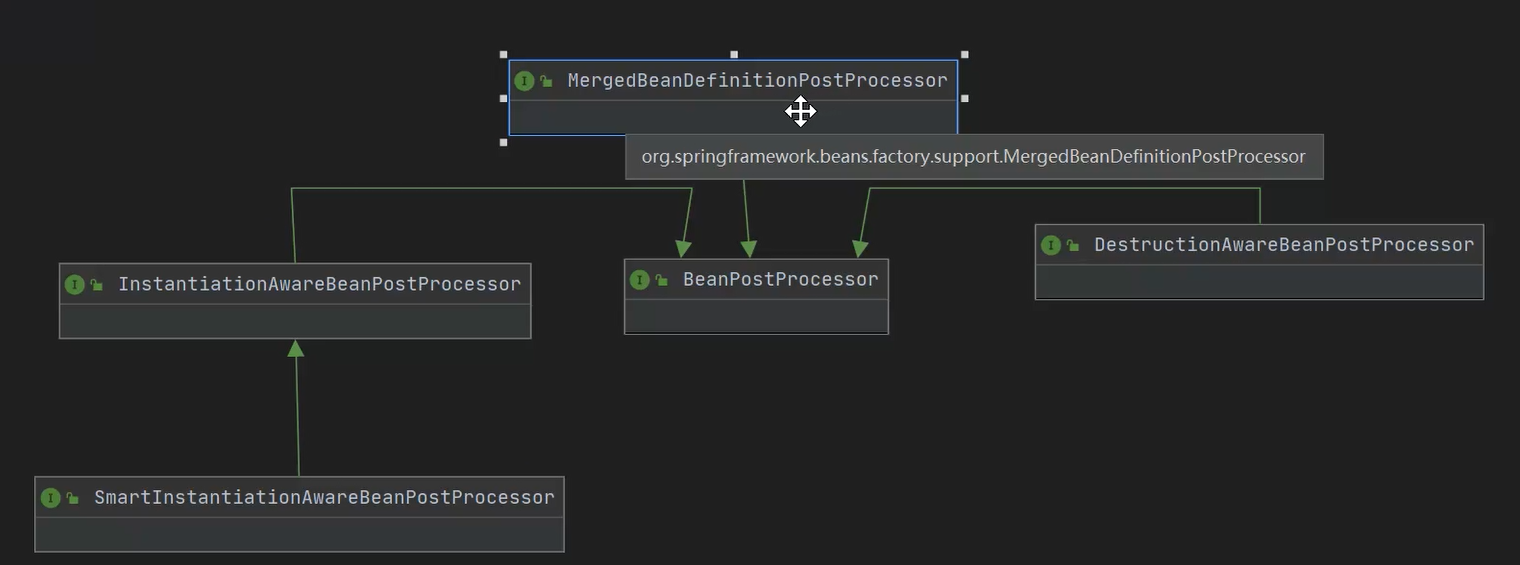
执行init-method

BeanPostProcessor:after 切面生成代理对象

Bean完成 -> context.getBean()







## springboot自动装配原理

1. spring启动在执行BeanFactory后置处理器ConfigurationClassPostProcessor,时会处理@Import注解,实例化引入类,并调用其实现方法
2. springboot启动注解中引入@Import(AutoConfigurationImportSelector),其逻辑,会加载META-INF/spring.factories文件,生产BeanDefinition

## BeanFactory 与 FactoryBean区别:

都是用来创建对象

BeanFactory由spring创建,遵守完整创建过程

FactoryBean用户实现接口,在getObject()中自己实现对象的创建,不走spring流程