Bean生命周期：创建—初始化-销毁

注意：对于单实例的bean，可以正常调用初始化和销毁方法

对于多实例，容器只负责初始化，但不会管理bean，容器关闭不会销毁bean

Bean的创建过程：

refresh—finishBeanFactoryInitialization--beanFactory.preInstantiateSingletons—getBean（去容器的缓存中拿）--doGetBean—createBean—doCreateBean—createBeanInstance（真正的产生了bean，并且包装了bean）--instantiateBean（返回一个无参构造bean）

bean的初始化

BeanPostProcessor原理:

可从容器类跟进顺序为:

AnnotationConfigApplicationContext-->refresh()-->

finishBeanFactoryInitialization(beanFactory)--->

beanFactory.preInstantiateSingletons()-->

760行getBean(beanName)--->

199行doGetBean(name, **null**, **null**, **false**)-->

317行createBean(beanName, mbd, args)-->

501行doCreateBean(beanName, mbdToUse, args)-->

541行createBeanInstance(beanName, mbd, args)(完成bean创建)-->

578行populateBean(beanName, mbd, instanceWrapper)(属性赋值)-->

579行initializeBean(beanName, exposedObject, mbd)(Bean初始化)->

1069行到1710行,后置处理器完成对init方法的前后处理.

最终得到如下如下

createBeanInstance(beanName, mbd, args)(完成bean创建)

populateBean(beanName, mbd, instanceWrapper); 给bean进行属性赋值

initializeBean() //初始化Bean方法内容如下,后置处理器对init方法的前后处理

{

applyBeanPostProcessorsBeforeInitialization(wrappedBean, beanName);

**invokeInitMethods(beanName, wrappedBean, mbd)** //执行自定义初始化

applyBeanPostProcessorsAfterInitialization(wrappedBean, beanName)

}

从以上分析不难发现,bean的生命周期为bean的创建, 初始化, 当容器关闭时对单实例的bean进行销毁.