

cachedIntrospectionResults：存放属性的PropertyDescriptor。

1. cachedIntrospectionResults初始化

cachedIntrospectionResults =CachedIntrospectionResults.forClass(getWrappedClass());

1. CachedIntrospectionResults构造方法
2. 属性描述器PropertyDescriptor(getter和setter)是如何拿取的？

通过java自带的Beaninfo接口获取this.beanInfo.getPropertyDescriptors()。

Spring创建了一个beanInfoFactory去获取beaninfo。工厂中拿不到beanInfo，就使用java自带的Introspector.getBeanInfo(Class clazz)去获取BeanInfo。

1. 每个bean在被实例化后，会返回一个BeanWrapper。

doCreateBean-----------》》》

instanceWrapper = createBeanInstance(beanName, mbd, args);

1. 每个beanWrapper都有一个cachedIntrospectionResults。cachedIntrospectionResults中有各个property的PropertyDescriptor。在获取PropertyDescriptor之后，会被封装成PropertyHandler。每个属性都对应一个PropertyHandler。PropertyHandler是AbstractNestablePropertyAccessor的内部类，而在BeanWrapper中有一个BeanPropertyHandler内部类继承了PropertyHandler。

FactoryBeanRegistrySupport--🡪>>>factoryBeanObjectCache记录