1.使用ClassPathXmlApplicationContext通过xml文件来创建实例

ApplicationContext xc = new ClassPathXmlApplicationContext("aa.xml");

2.通过文件路径加载配置文件

setConfigLocations(configLocations);

3.方法入口 refresh

refresh();

1.prepareRefresh()
2.ConfigurableListableBeanFactory beanFactory = obtainFreshBeanFactory()
3.prepareBeanFactory(beanFactory)
4.postProcessBeanFactory(beanFactory)
5.invokeBeanFactoryPostProcessors(beanFactory)
6.registerBeanPostProcessors(beanFactory)
7.initMessageSource();
8.initApplicationEventMulticaster()
9.onRefresh()
10.registerListeners()
11.执行类的实例化和初始化操作
finishBeanFactoryInitialization(beanFactory)
12.finishRefresh()
13.resetCommonCaches()

4.refresh()方法内部包含13个方法

5.执行类的实例化和初始化操作

finishBeanFactoryInitialization(beanFactory)

for(1.循环beanNames逐个实例化和初始化)
if(!(

2.判断是否是抽象类/单利/懒加载
3.判断是否继承FactoryBean
) else {

4.//这里开始实例化
getBean(beanName);
}

6.初始化不是懒加载的单例

beanFactory.preInstantiateSingletons();

7.1 getBean(beanName)
7.2 getBean(beanName)

8. 一级缓存查找失败
Object sharedInstance = getSingleton(beanName);

markBeanAsCreated(beanName);

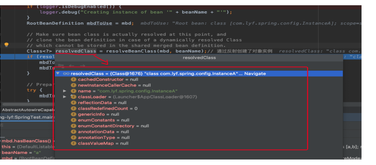
public Object getSingleton(String beanName, ObjectFactory<?> singletonFactory)

singletonObject = singletonFactory.getObject();

创建InstantB

这里通过调用singletonFactory.getObject()来调用匿名内部类

createBean(String beanName, RootBeanDefinition mbd, @Nullable Object[] args)



Class<?> resolvedClass = resolveBeanClass(mbd, beanName);// 通过反射创建了对象实例

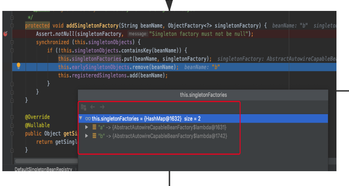
Object beanInstance = doCreateBean(beanName, mbdToUse, args);

addSingletonFactory(beanName, () -> getEarlyBeanReference(beanName, mbd, bean));

populateBean(beanName, mbd, instanceWrapper);//填充bean

填充的时候发现需要实例化InstantB

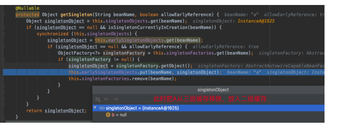
//这里去实例化InstantB
Object resolvedValue = valueResolver.resolveValueIfNecessary(pv, originalValue);



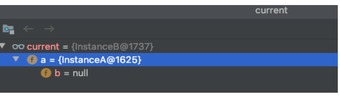
和之前的A去找B相比差别就是，A对象现在处于创建中，所以可以进入if条件语句中

同理InstanceB去缓存中去找A是否存在

B已经在一级缓存中了，再来处理A



把A从三级缓存移除，放入二级缓存，并返回A对象的实例

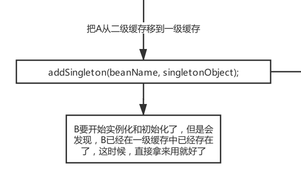


中间经过一系列操作，开始初始化B

exposedObject = initializeBean(beanName, exposedObject, mbd);

直接把B从三级缓存引进了一级缓存？？

addSingleton(beanName, singletonObject);



把A从二级缓存移到一级缓存

addSingleton(beanName, singletonObject);

到这里我们A的实例化和初始化都完成了

B要开始实例化和初始化了，但是会发现，B已经在一级缓存中已经存在了，这时候，直接拿来用就好了