有道云链接：http://note.youdao.com/noteshare?id=1f3f25f6dc56c043ce8cc9dca586f2db&sub=7C4CF0F86D134F4BB21460D8F0271A42

Bean的销毁过程

Bean销毁是发送在Spring容器关闭过程中的。

在Spring容器关闭时，比如：

```
AnnotationConfigApplicationContext context = new
AnnotationConfigApplicationContext(AppConfig.class);
UserService userService = (UserService) context.getBean("userService");
userService.test();

// 容器关闭
context.close();
```

在Bean创建过程中，在最后（初始化之后），有一个步骤会去判断当前创建的Bean是不是DisposableBean：

1. 当前Bean是否实现了DisposableBean接口
2. 或者，当前Bean是否实现了AutoCloseable接口
3. BeanDefinition中是否指定了destroyMethod
4. 调用DestructionAwareBeanPostProcessor.requiresDestruction(bean)进行判断
    i. ApplicationListenerDetector中直接使得ApplicationListener是DisposableBean
    ii. InitDestroyAnnotationBeanPostProcessor中使得拥有@PreDestroy注解了的方法就是DisposableBean
5. 把符合上述任意一个条件的Bean适配成DisposableBeanAdapter对象，并存入disposableBeans中（一个LinkedHashMap）

在Spring容器关闭过程时：

1. 首先发布ContextClosedEvent事件
2. 调用lifecycleProcessor的onCloese()方法
3. 销毁单例Bean
    i. 遍历disposableBeans
        a. 把每个disposableBean从单例池中移除
        b. 调用disposableBean的destroy()
        c. 如果这个disposableBean还被其他Bean依赖了，那么也得销毁其他Bean
        d. 如果这个disposableBean还包含了inner beans，将这些Bean从单例池中移除掉(inner bean参考https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/core.html#beans-inner-beans)
    ii. 清空manualSingletonNames，是一个Set，存的是用户手动注册的单例Bean的beanName

iii. 清空allBeanNamesByType，是一个Map，key是bean类型，value是该类型所有的beanName数组
iv. 清空singletonBeanNamesByType，和allBeanNamesByType类似，只不过只存了单例Bean

这里涉及到一个设计模式：**适配器模式**

在销毁时，Spring会找出实现了DisposableBean接口的Bean。

但是我们在定义一个Bean时，如果这个Bean实现了DisposableBean接口，或者实现了AutoCloseable接口，或者在BeanDefinition中指定了destroyMethodName，那么这个Bean都属于“DisposableBean”，这些Bean在容器关闭时都要调用相应的销毁方法。

所以，这里就需要进行适配，将实现了DisposableBean接口、或者AutoCloseable接口等适配成实现了DisposableBean接口，所以就用到了DisposableBeanAdapter。

会把实现了AutoCloseable接口的类封装成DisposableBeanAdapter，而DisposableBeanAdapter实现了DisposableBean接口。