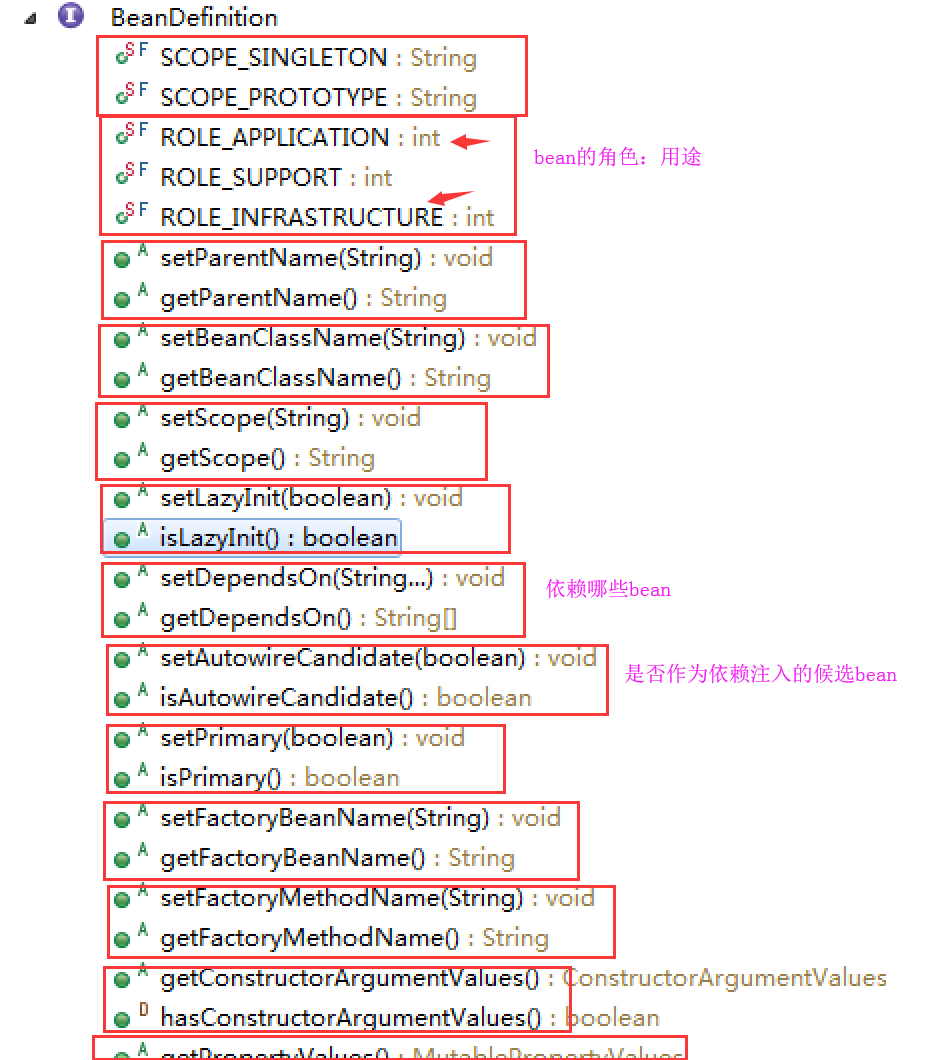
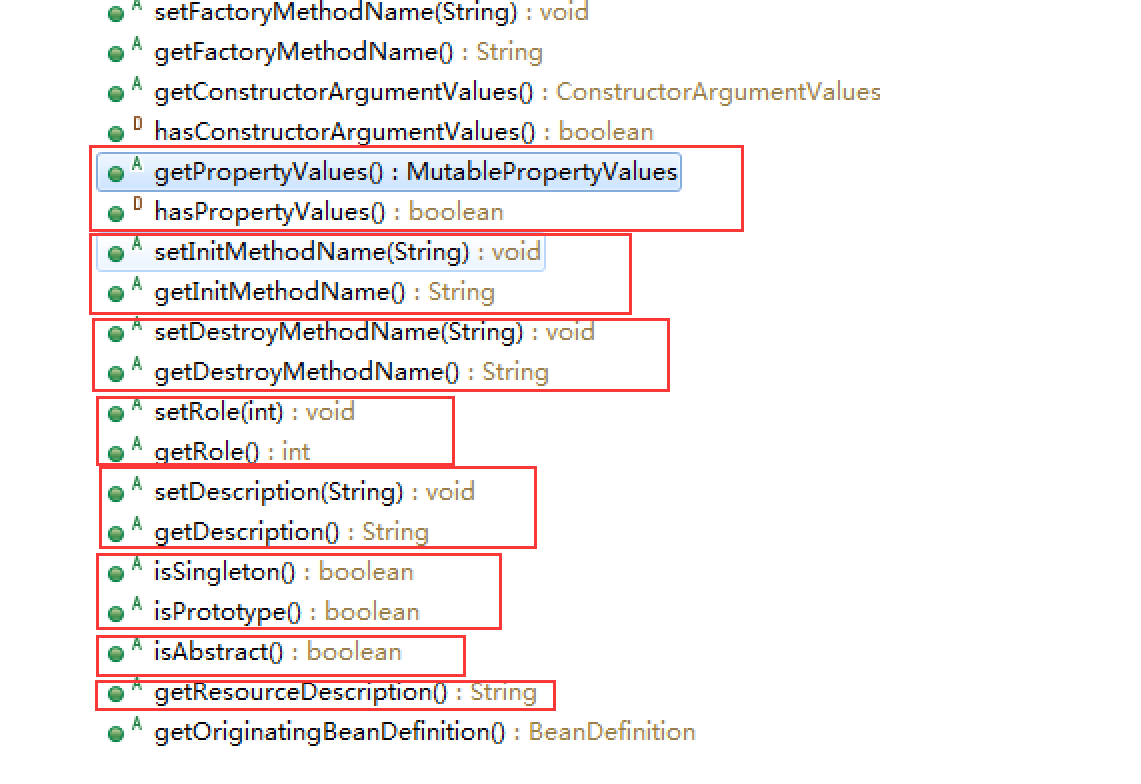
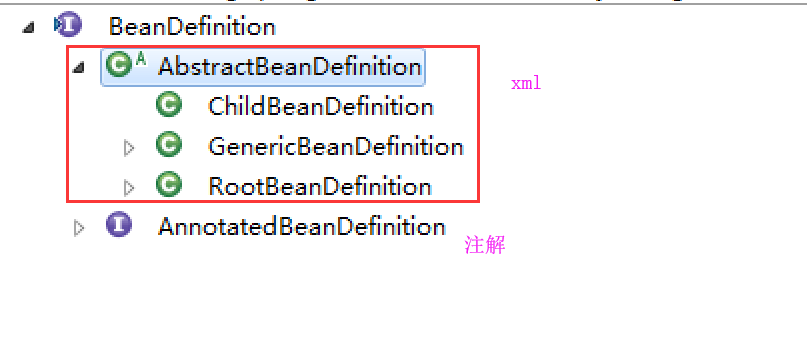
# Spring 源码学习

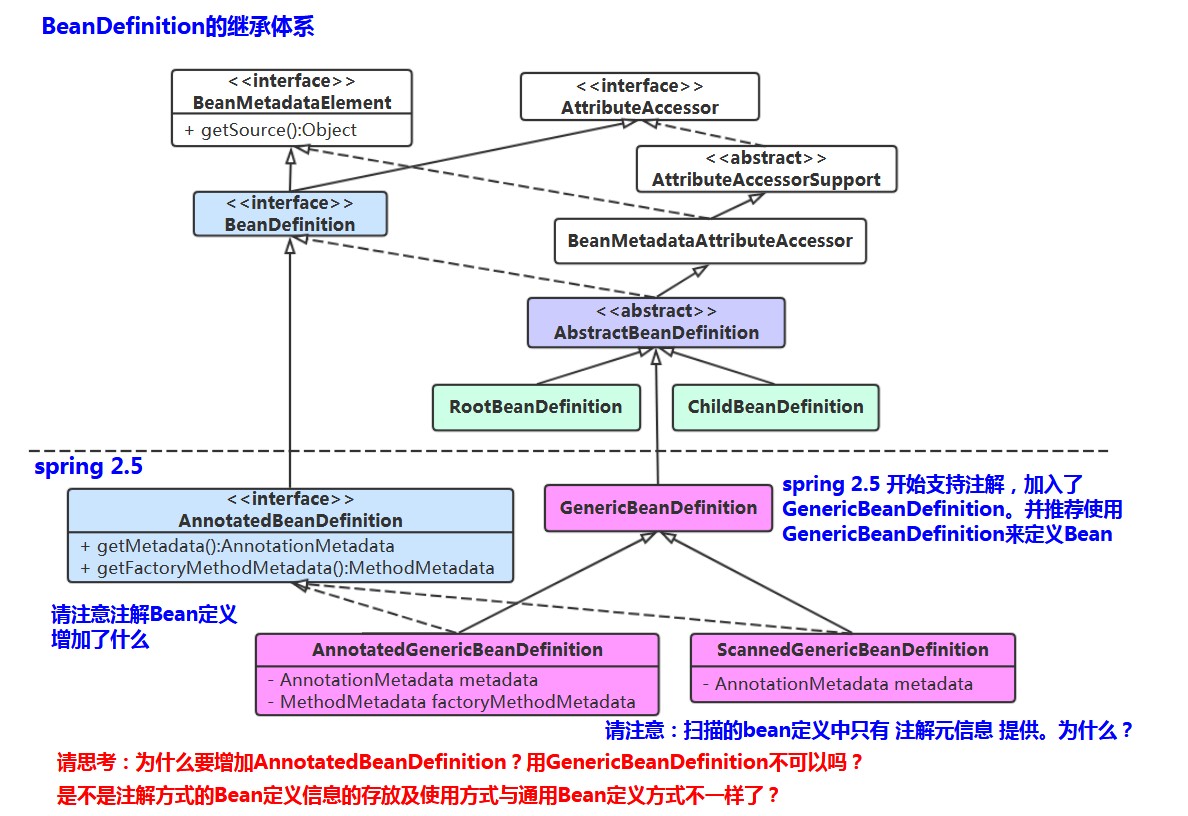
## BeanDefinition

1、定义了什么



2、BeanDefinition 的继承体系

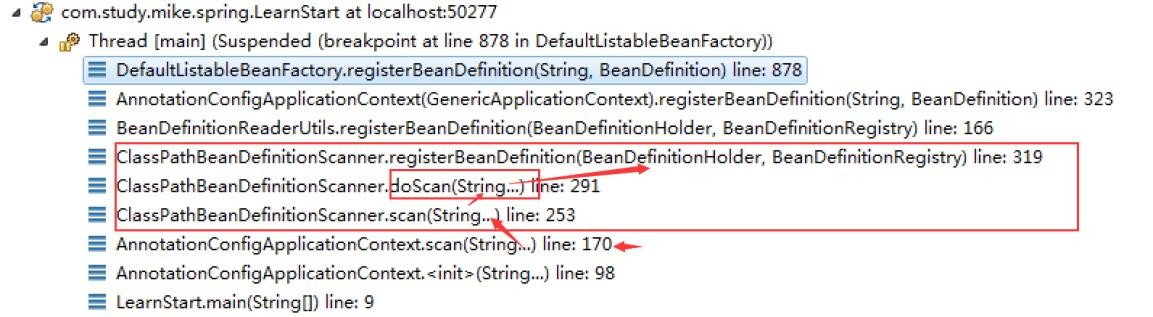




## Annotation 方式配置的的 BeanDefinition 的解析

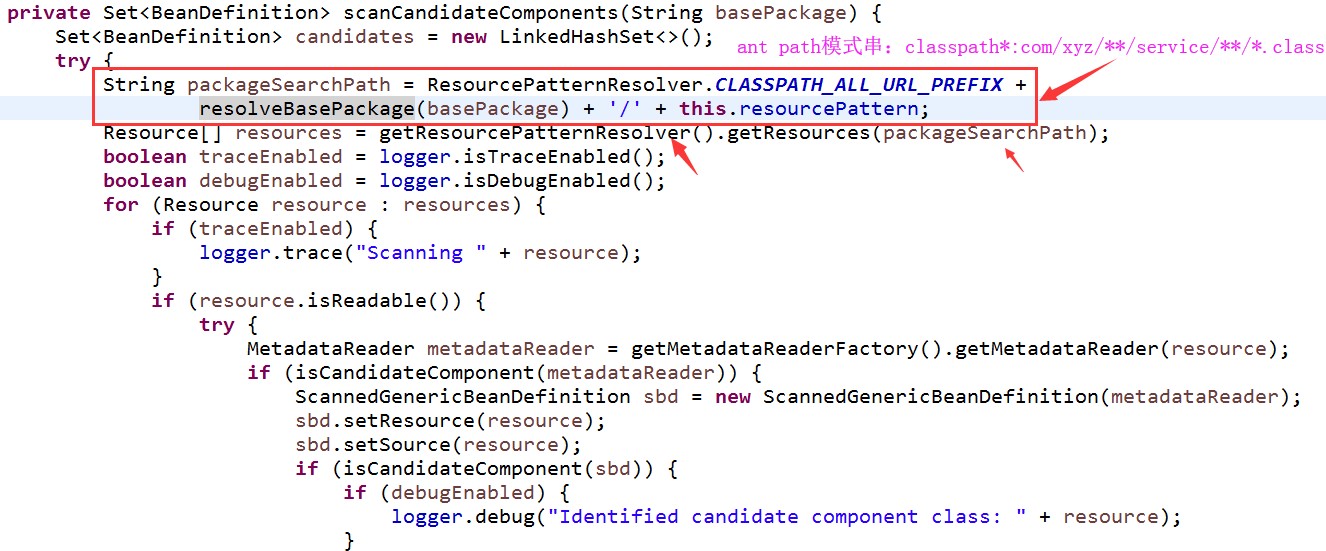
* + 1. **扫描的过程，如何扫描**

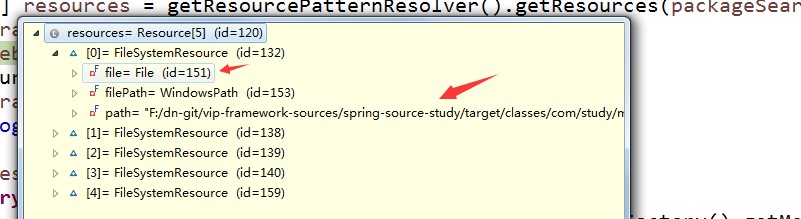
怎么去看的，断点





同过 ant path 模式匹配来扫描包下的.class 文件





## 注解的解析

1、 如何从扫到的.class 文件中获得注解信息？ 我们自己实现时是如何来做的：

1、 Class.forname(“className”)加载类获得 Class 对象

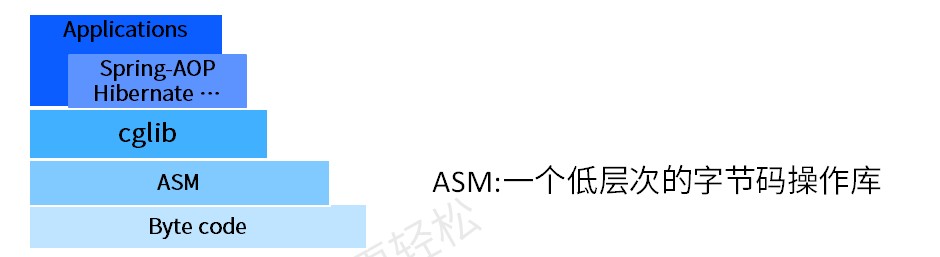
2、 反射获取注解

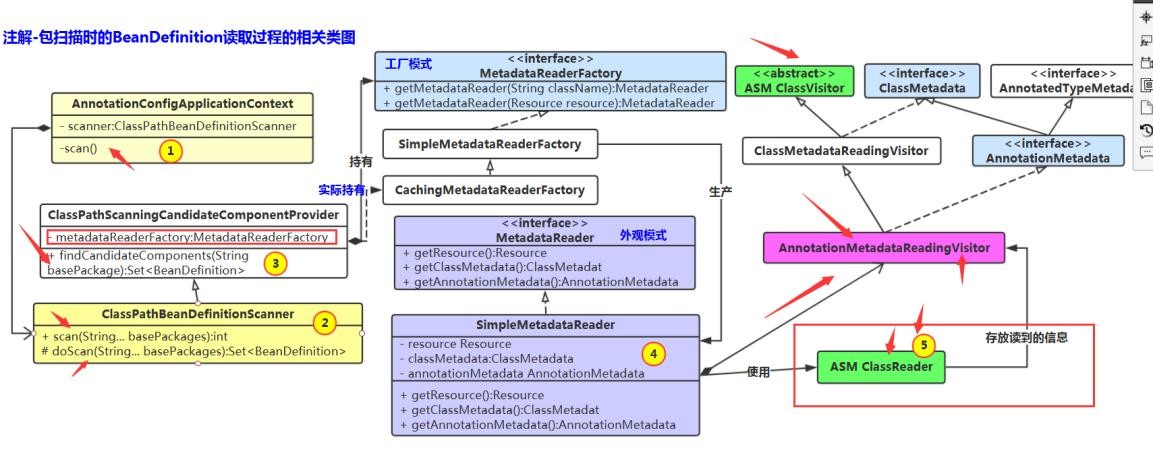
3、 判断是否存在组件注解，存在为其创建 BeanDefinition

4、 看指定了名字没，如没，应用名字生成策略生成一个名字

5、 注册 BeanDefinition

2、 Spring 的解析过程：





3、 提问：spring 中通过 ASM 字节码操作库来读取的类信息、注解信息。它没有加载类，为什么不用加载类的方式？。

4、 MetadataReader 读取到类信息、注解信息后，如何进行判断及创建 BeanDefinition 的， 往 BeanDefintion 中给入了哪些信息。

5、 请思考，我们可以自己定义标注组件的注解吗？【扩展点】

https://docs.spring.io/spring/docs/5.1.3.RELEASE/spring-framework-reference/core.html#be ans-meta-annotations

6、 扫描的过滤这块你在实际项目中是否用过？如何使用的？【扩展点】

https://docs.spring.io/spring/docs/5.1.3.RELEASE/spring-framework-reference/core.html#beans- scanning-filters

