由于Spring的注释写的比较好，我可以很方便的从Spring的注释中获取很多信息，先从调试进去后看到第一个接口：

PropertyResolver：注释只说了对某些底层的properties解析，接口本身对看源码没有意义

Environment：注释写了很多信息，首先知道Spring支持Profile环境配置（比如可以自定义开发环境线上环境等等），然后properties是由EnvironmentAware接口管理的，包括经常用的${...}方式的properties也是其中的实现之一。

ConfigurableEnvironment：注释说明可以根据MutablePropertySources来添加移除替换PropertySource，而且可以设置激活哪个Profile

AbstractEnvironment：提供了一些properties中的用来指定活跃的Profile或者默认Profile的名称key

AbstractRefreshableConfigApplicationContext#setConfigLocations：循环处理传入的配置文件路径，使用AbstractPropertyResolver#doResolvePlaceholders方法然后替换传入配置文件路径中的替换字符，比如spring-${evn}.xml替换成在AbstractEnvironment中存在的properties的值。

AbstractApplicationContext#refresh：

1. 使用一个对象进行加锁
2. 通过一个ApplicationStartup来创建一个性能监控类，使用默认实现就是什么也不做，Spring还实现了一个Java Flight Recorder的实现类，也不知道咋用。
3. prepareRefresh方法刷新上下文bean之前的准备工作，主要是
4. initPropertySources初始化配置的properties，ClassPathXmlApplicationContext并没有实现这个方法，但是在web包中的一些ApplicationContext实现了这个方法，使用ServletContextPropertySource来加载web.xml中配置的properties文件。
5. getEnvironment().validateRequiredProperties()验证一些必要的properties是否存在。
6. 设置需要的ApplicationListener

DefaultListableBeanFactory#setAllowBeanDefinitionOverriding：是否允许替换同名的bean，默认是允许

DefaultListableBeanFactory#setAllowCircularReferences：是否允许循环引用，默认是允许，如果不允许则有循环引用就会报错