

作业

写出阅读源码的注意事项（至少五个）

- 不要通读源码，要带着目的与期望阅读源码。
- 明确主线，对于旁支一眼带过或者作个标记回头再详细阅读
- 找到入口，利用DEBUG 工具，这样可以很清晰的看到运行时栈信息
- 明确层次，在主流程没有弄清楚之前不要去想设计模式应用的相关问题
- 做好注释。以免看懂之后回头又忘记
- 做好笔记，多次阅读

写出getBean流程中相关的类及作用（至少6个）

- `BeanFactory` 接口
 - 直接定义了 `getBean` 方法，以及相关的重载方法。可以通过各种接口方法获取到相应的 bean 实例
- `ListableBeanFactory` 接口
 - 继承 `BeanFactory` 接口，并扩展了 Bean 对象是 List 列表形式的 `getBean` 接口方法
- `AutowireCapableBeanFactory` 接口
 - 继承 `BeanFactory` 接口，并扩展了 `createBean` 相关的接口方法
- `AbstractBeanFactory` 抽象类
 - `getBean` 方法的主要实现步骤都在这个抽象类中实现。
- `AbstractAutowireCapableBeanFactory`
 - 实现了 `createBean` 方法。
- `DefaultListableBeanFactory` 实现类
 - 所有父类没有实现的方法，都在这里实现。
- `SingletonBeanRegistry` 接口
 - 单例 Bean 实例的工厂接口类，主要定义了单例 Bean 对象的注册以及获取接口方法

写出AOP的核心概念（至少6个）

- 什么是AOP
 - 直接翻译过来就是面向切面编程
- 连接点(`Joinpoint`)
 - 连接点就是指的那些被拦截到的点。
 - 在spring 中就是通过一定的方法将一些方法在调用时截取下来。添加一些自定义的逻辑
- 切入点(`Pointcut`)
 - 切入点就是我们要拦截的点。

我们要对哪些方法、类进行拦截，那么这些方法或者类就是我们的切入点。

- 通知/增强(Advice)

就是被拦截到，然后添加一些新的逻辑，这些逻辑我们就把它叫做通知。

通知分为前置通知、后置通知、异常通知、环绕通知等。

- 代理(Proxy)

一个类被AOP织入增强后所产生的新的结果类，用来代替原来的类。

- 切面(Aspect)

切入点和通知的结合

- 目标对象(Target)

被拦截的类对象

- 织入(Weaving)

是指把增强应用到目标对象来创建新的代理对象的过程