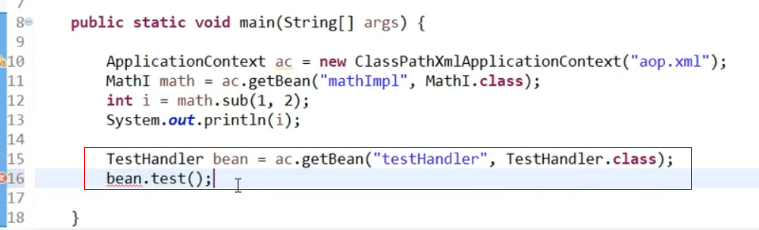
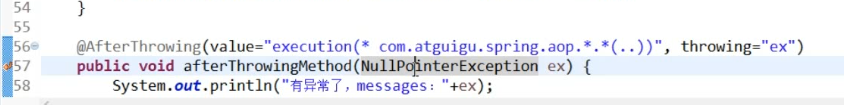
* **Spring和AspectJ结合使用实现AOP机制**

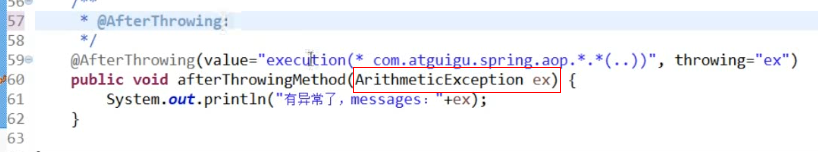
1. Spring最核心的还是IOC，
2. 即使AOP也要靠IOC来实现，即动态代理作用的目标对象必须由Spring 来管理，因此要加@Component等注解
3. Spring框架中，如果目标类没有接口，使用AOP时，执行的是cglib，实际这个类TestHandler.java继承的是Object.java

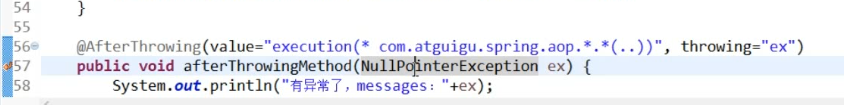




(4)返回通知@AfterReturing如果有异常则不会执行，其中returning属性用来接收目标类方法（本例中add(), plus()等等)的返回值，接收后变为Object类型，注意returning=”result”中的属性值result要和形参中Object result变量名要一致，否则报错

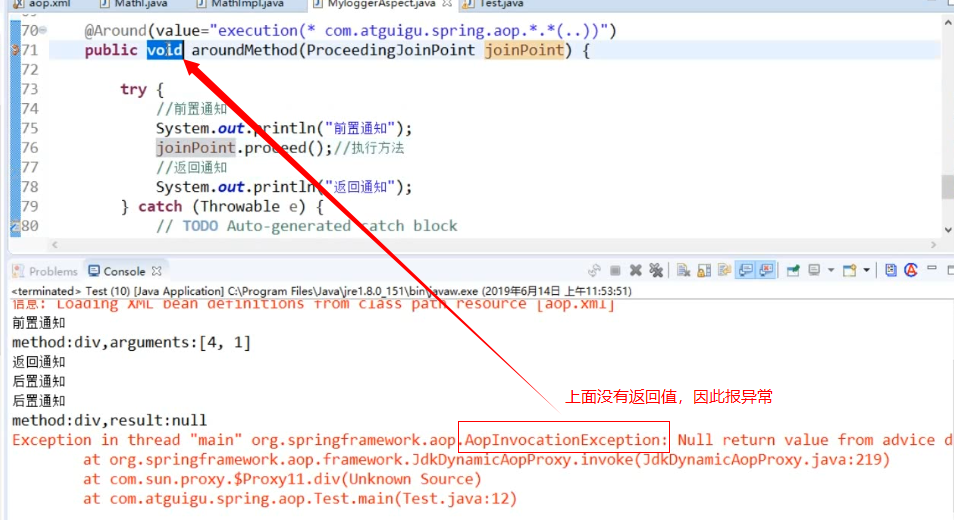


1. 异常通知@AfterThrowing ，如果方法中形参类型改为（Exception ex）则接收所有目标类方法中的异常并显示通知，如果不是目标类方法抛出的异常则不打印通知，例如Arithmeticexception ex是正确的异常(bt zero)，若改为NullPointerException ex则不打印任何通知，throwing属性就是用来接收异常的

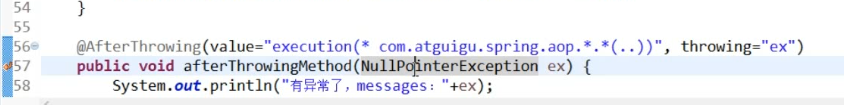


1. 环绕通知，用的较少，实际跟动态代理的代码写法类似

返回值类型可改为Object



* (1)切入点表达式：即@Before,@After等后面括号内的表达式，形参内两点“..”表示

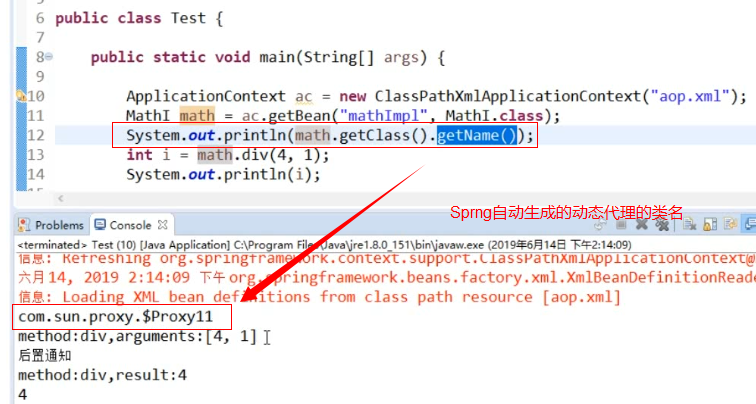


(2)切入点用法，也叫重用切入点方式



* 切面就是AOP程序，动态代理类，如日志功能的java程序



* Spring生成的动态代理类名
* 如果一个目标类有两个切面AOP的动态代理类，默认执行先加载的动态代理，也可设置优先级，使用@Order(1)，值越小优先级越高



* 注意引入pointcut标签后属性名不同

