@Component

被该注解标识的类会在指定扫描范围内被Spring加载并实例化。



由于JavaEE开发是分层的，@Component又衍生出如下三个注解：

@Repository在Dao类上使用

@Service在Service层类上使用

@Controller在Web层类上使用



@Autowired根据类型进行注入，如果同一类型的Bean有多个，尝试根据名字进行二次匹配，匹配不成功报错。

@Resource不指定名称参数时，根据类型注入，指定名称就根据名称注入。

@Autowired

public void xxx(UserDao userDao){

System.out.println(“xxx:”+userDao);

}

先去找UserDao类型，再去找名称为userDao的，注入进去

@Autowired

Public void yyy(List<UserDao> userDaoList){

System.out.println(“yyy:”+userDaoList);

}

先去找UserDao类型，找到了多个就依次注入进list中。





















AOP常見用法

性能监控：在方法调用前后记录调用时间，方法执行太长或超时报警

缓存代理：缓存某方法的返回值，下次执行该方法时，直接从缓存里获取

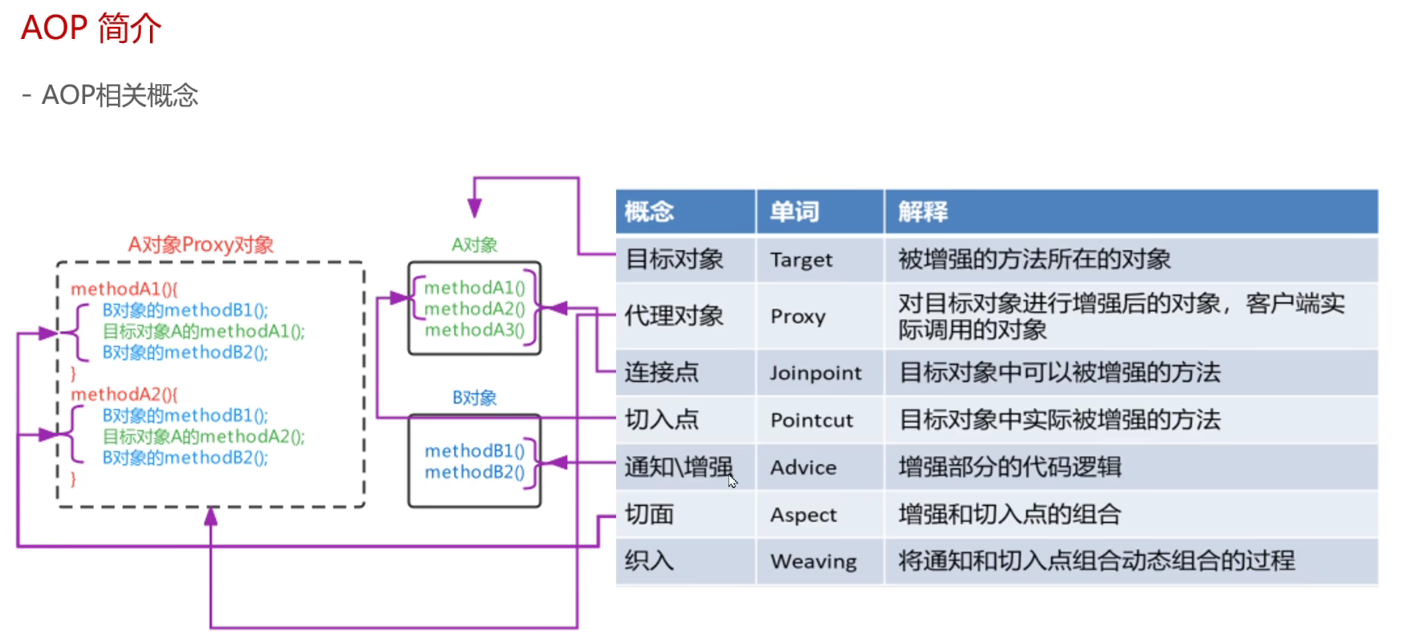
软件破解：使用AOP修改软件的验证类的判断逻辑

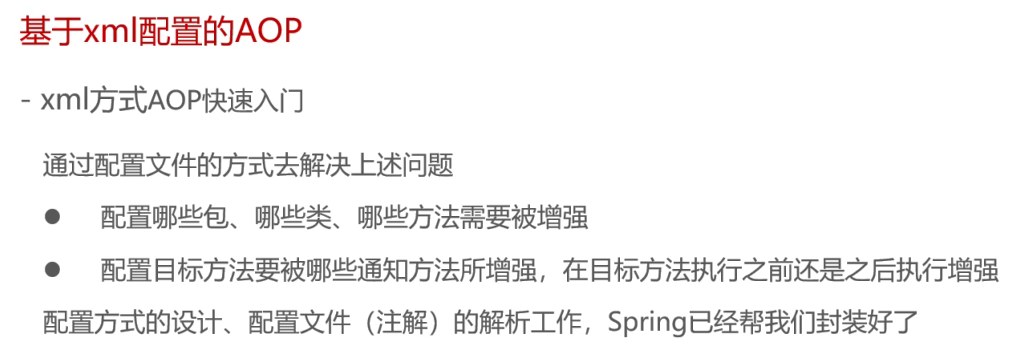
记录日志：在方法执行前后记录系统日志

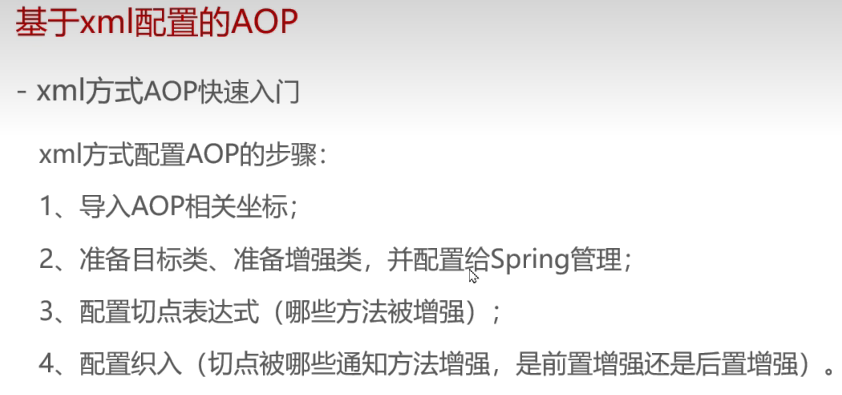
工作流系统：工作流系统需要将业务代码和流程引擎代码混合在一起执行，那么我们可以使用AOP将其分离，并动态挂接业务

权限验证：方法执行前验证是否有权限执行当前方法，没有则抛出没有权限执行异常，由业务代码捕捉



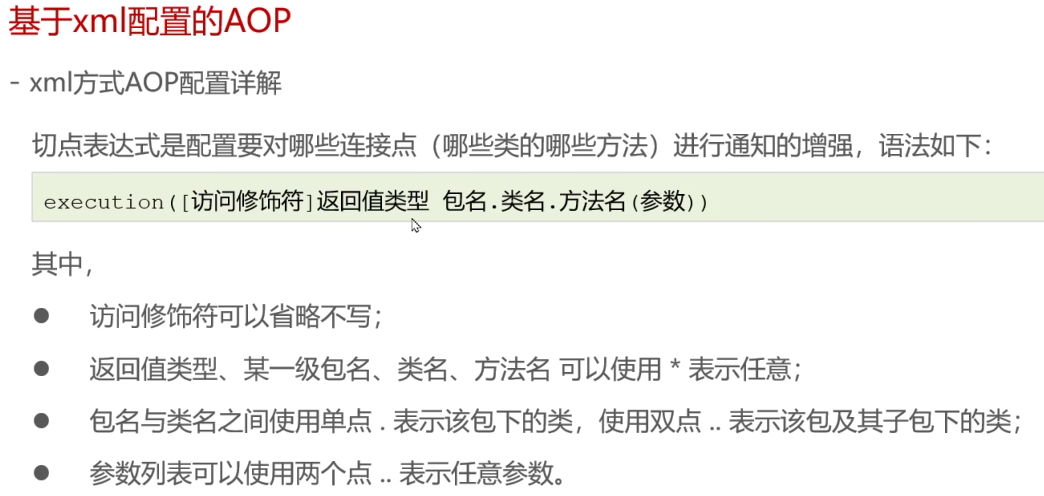






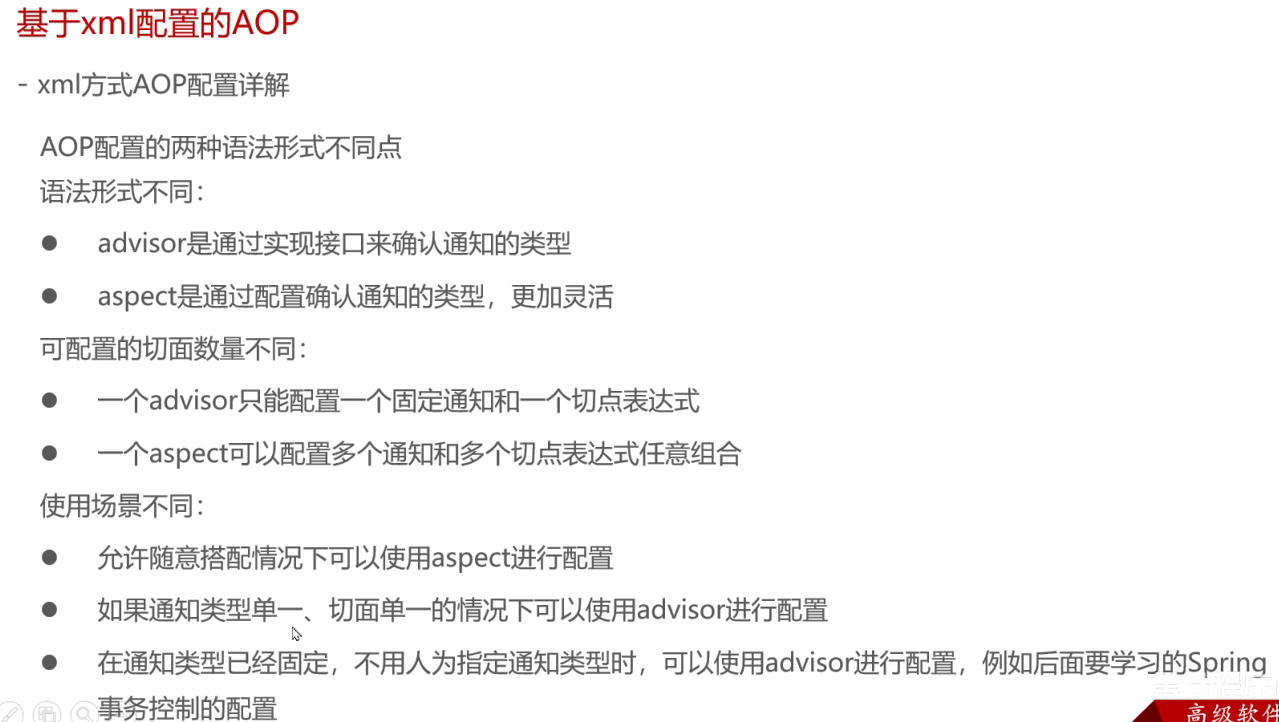






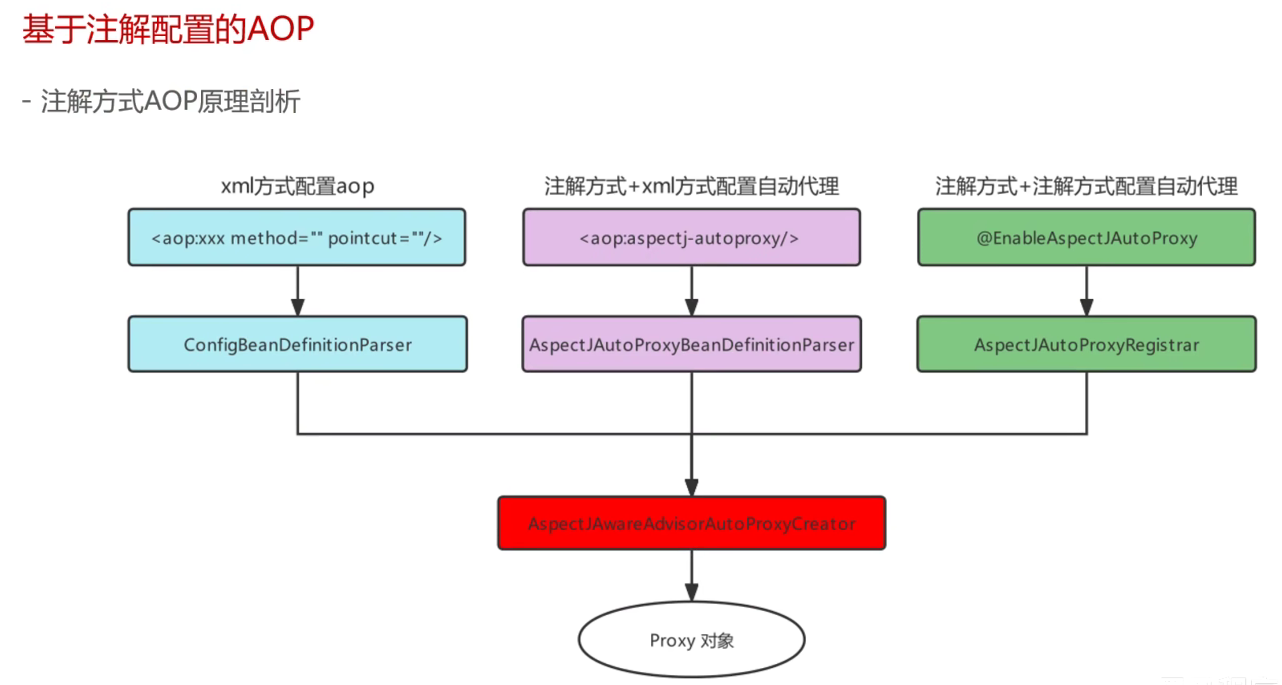


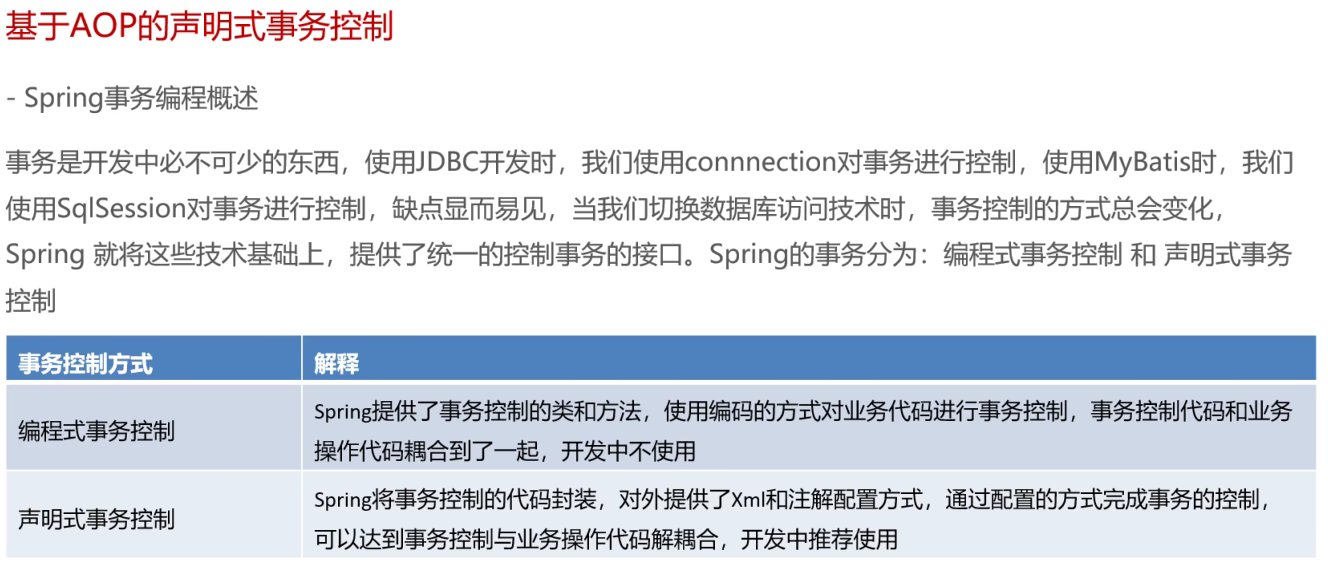


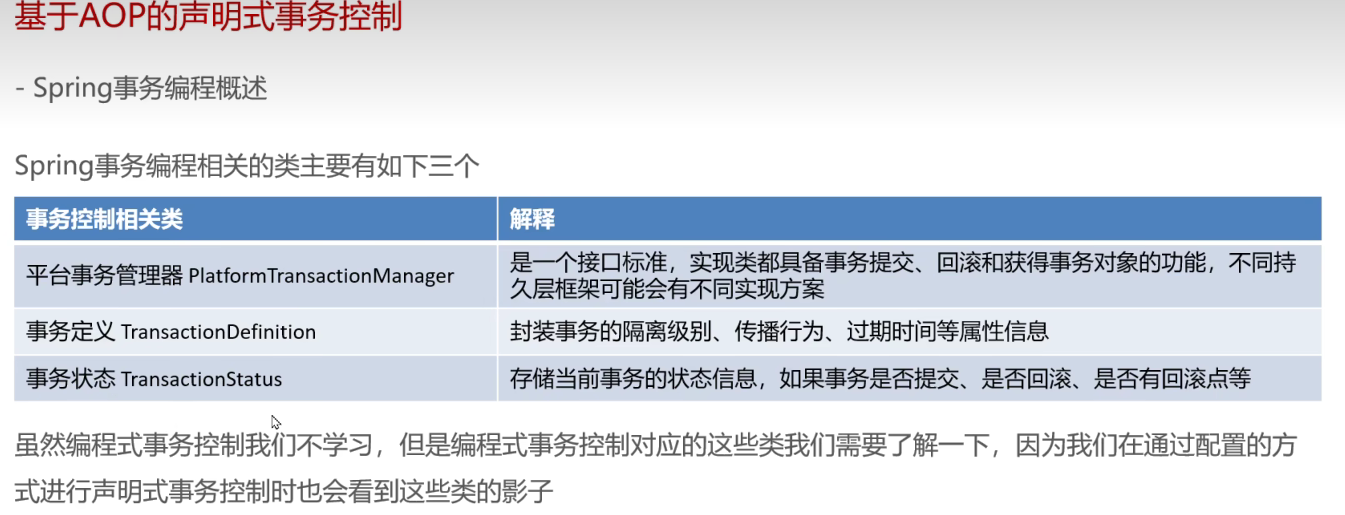


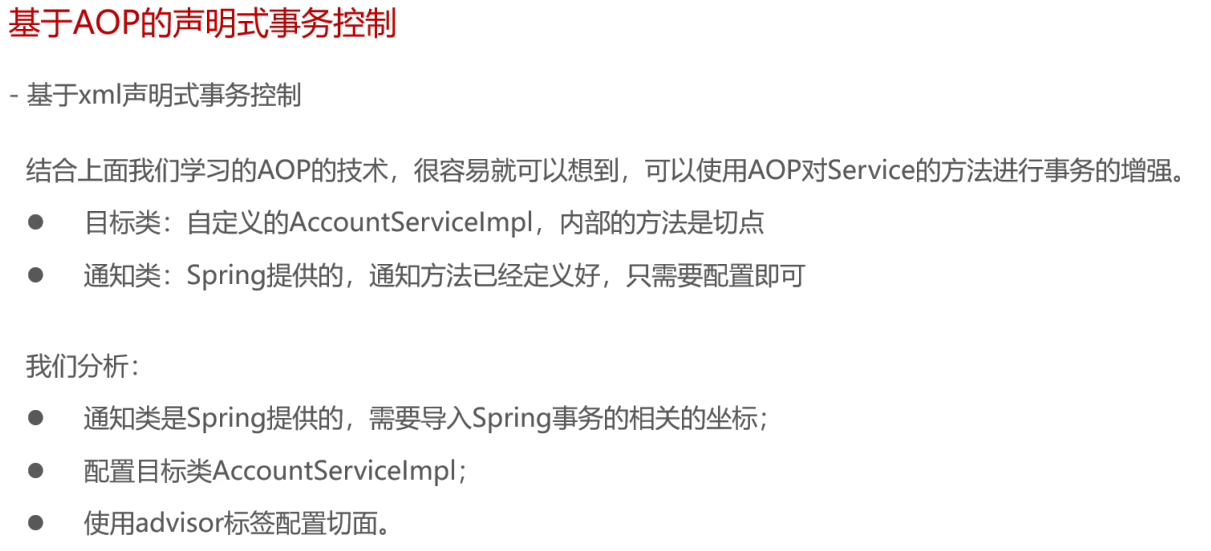
Advisor開發的時候基本不配，看看就行











事务的隔离级别

脏读（读取了未提交的新事务，然后被回滚了）：

事务A读取了事务B中尚未提交的数据。如果事务B回滚，A读取使用了错误的数据。

不可重复读（读取了提交的新事务，指更新操作）：

不可重复度是指在对于数据库中的某个数据，一个事务范围内多次查询却返回了不同的数据值，这是由于在查询间隙数据被另一个事务修改并提交了。

幻读（也是读取了提交的新事务，指增删操作）：

在事务A多次读取过程中，事务B对数据进行了新增或者修改操作，导致事务A多次读取的数据不一致。

Read uncommitted：一个事务可以读取另外一个未提交事务的数据，会脏读，会产生不可重复读，幻读

Read committed：一个事务要等另一个事务提交以后才能读取数据，会产生不可重复读，幻读

Repeatable read：重复读就是在事务开启的时候，不允许修改，可能会幻读。

Serializable：事务串行化顺序执行，避免脏读幻读不可重复读。效率低下，比较消耗数据库性能一般不使用。

大多数数据库默认的隔离级别是Read committed，如sql server ，oracle

Mysql默认隔离级别是Repeatable read

事务的传播机制

https://zhuanlan.zhihu.com/p/148504094

**REQUIRED(Spring默认的事务传播类型)**

**如果当前没有事务，则自己新建一个事务，如果当前存在事务，则加入这个事务（个人理解，如果外层没有事务，则自己作为一个新事务，如果外层有事务，就加入这个事务）**

事务失效原因

https://zhuanlan.zhihu.com/p/334766548