**AOP术语：**

Joinpoint（连接点）：可以插入增强代码的时机点；（方法开始时、方法结束时、正常运 行完毕时、运行出现异常时等时机点）

Pointcut（切入点）：需要执行增强代码的方法

Advice（增强）：

层次一：指的是增强代码逻辑；

层次二：方位点（在部分连接点上加入增强代码，这些加入增强代码的连接点就是 方位点；也就是方法中的需要添加增强代码的连接点）；

Target（目标对象）：被代理的对象（即需要增强的方法所在的对象）；

Proxy（代理对象）：方法被增强后产生的对象；

Weaving（织入）：增强原有方法并产生代理对象的这个过程；

Aspect（切面）：增强代码所关注的各方面，存放相关增强代码的类称之为切面类，

如：事务切面，其中定义的代码都是与事务相关的代码，包括：开启事务、提交事 务、回滚事务等；

**Spring声明式事务的支持**

编程式事务：在业务代码中使用try-catch控制事务，即：业务代码和事务控制代码耦 合在一起，这样的事务控制机制就叫编程式事务。

声明式事务：通过xml或注解的方式实现事务控制的机制，就是声明式事务。

事务的四大特性：

原子性：事务是一个不可拆分的单位。从操作的角度来看，事务中的操作要么都执 行，要么都不执行。

一致性：从数据角度来看，事务中的数据必须从一个一致状态转变为另一个一致状 态。

隔离性：未被提交的事务不应该被其他事务读取。

持久性：事务一旦被提交，它对数据库中的数据的改变是永久的。

隔离级别在并发时会出现的问题：

脏读：读取未提交事务的数据；

不可重复读：同一条数据update前后读取到的信息不一致；

幻读：insert、delete数据会出现两次相同查询但结果却不一致的情况；

事务的隔离级别：

Serializable（串行化）：避免脏读、不可重复读、幻读的发生；

Repeatable read（可重复读）：避免脏读、不可重复读，幻读有可能发生；

对update的数据加锁

Read Committed（读已提交）：避免脏读，不可重复读和幻读一定会发生；

Read Uncommitted（读未提交）：三种情况都无法避免；

事务的传播行为：

若A、B皆已添加事务控制，A中调用了B时，就需要进行事务的协商，这个过程就叫做事务的传播行为。

一般情况下：若当前A没有事务就新创建一个事务；若当前A有事务B就加入到A的事务中。