注意一个问题，spring将事务的管理委托给了持久层的框架。

一开始的时候是jdbc的事务管理。

Spring的事务管理spi的抽象层主要包括3个接口，分别是PlatformTransationManager,TransationDefinition和TransationStatus.

编程式事务管理，

使用xml配置声明式事务管理(在xml中配置属性，麻烦)

使用注解配置声明式事务：

<tx:annotation-driven transaction-manager="transactionManager" />

<bean id=*"transactionManager"*

class=*"org.springframework.jdbc.*

*datasource.DataSourceTransactionManager"*>

<property name=*"dataSource"* ref=*"dataSource"* />

</bean>

关于@Transactional的属性：他拥有一组普适性很强的事务属性，我们直接使用默认的配置就可以

包括几点：



添加过@Transactional的方法就可以在事务环境下工作。

事务管理是为了保证数据操作的事务性(原子性，一致性，隔离性，持久性即所谓的ACID)，脱离了事务性，DAO照样可以进行数据的操作。

Spring框架提供的各种的好处的唯一前提就是让pojo的类变成受Spring容器管理的Bean。另外，一点就是，在项目开发中不需要有一种固化思维:controller,service,serviceimpl,dao,.xml这样的分层思想，面向接口编程，有时是可以变通一下的。

事务的传播行为就是多个事务之间相互调用时，事务如何在这些方法中间传播。