一，事务的概念

1，什么是事务？

事务是一系列的操作的集合。

2，事务的特性：原子性、一致性、隔离性、持久性。

3，不考虑隔离性产生脏读、不可重复读、幻读（虚读）。可以通过设置事务的隔离性级别来解决这个问题。

二，spring中的事物管理

1，spring事务管理api

（1）spring事务管理俩种方式

第一种：编程式事务管理（不用）。

第二种：声明式事务管理

①基于xml配置文件实现

②基于注解方式实现

三，spring进行事务的api介绍

1，spring事务管理高层抽象主要包括3个接口。

2，platformTransactionManager事务管理

3，TransactionDefinition事务定义信息（隔离、传播、超时、只读）

4，Transaction事务具体运行状态。

5，spring针对不同的dao层框架，提供接口不同的实现类。

6，无论使用注解方式还是xml文件方式，都需要首先配置事务管理器

四，搭建转账环境

1，创建数据库表，添加数据

```
create table user(id int primary key auto_increment,username varchar(20),salary int);
insert into user values(1,'anqili',1000),(2,'linmingjun',1000);
```

2，创建一个service（也叫业务逻辑层）和dao类，完成注入关系

（1）service层又叫业务逻辑层。

（2）dao层单纯对数据库操作层，在dao层不写任何业务。

（3）需求：anqili转给linmingjun100元钱。

五，没有采取事务操作的代码

1，orderDao实体类

```
package cn.java.ZhuanZhang.Dao;
import org.springframework.jdbc.core.JdbcTemplate;
public class OrderDao {
	//注入jdbctemplate
private JdbcTemplate jdbcTemplate;
public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {
	this.jdbcTemplate = jdbcTemplate;
}
/**
 * 做对数据库操作的方法，不写业务操作
 */
//=少钱的方法
public void lessMoney(){
	String string = "update user set salary = salary-?  where username=?";
	jdbcTemplate.update(string,100,"anqili");
}
//多钱的方法
public void moreMoney(){
	String string = "update user set salary = salary+?  where username=?";
	jdbcTemplate.update(string,100,"linmingjun");
}
}
```

2，orderService实体类

```
package cn.java.ZhuanZhang.Service;
import cn.java.ZhuanZhang.Dao.OrderDao;
public class OrderService {
private OrderDao orderDao;
public void setOrderDao(OrderDao orderDao) {
	this.orderDao = orderDao;
}
//调用dao实体类中的方法
```

```
//业务逻辑层
public void accountMoney(){
	//anqili少100元
	orderDao.lessMoney();
	//linmingjun多100元
	orderDao.moreMoney();
}
}
```

3，bean3.xml配置文件

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
	xmlns:jdbc="http://www.springframework.org/schema/jdbc"
xmlns:jee="http://www.springframework.org/schema/jee"
	xmlns:tx="http://www.springframework.org/schema/tx"
xmlns:util="http://www.springframework.org/schema/util"
	xmlns:task="http://www.springframework.org/schema/task"
	xsi:schemaLocation="
	http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
	http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd
	http://www.springframework.org/schema/jdbc
http://www.springframework.org/schema/jdbc/spring-jdbc-4.0.xsd
	http://www.springframework.org/schema/jee
http://www.springframework.org/schema/jee/spring-jee-4.0.xsd
	http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
	http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util-4.0.xsd
	http://www.springframework.org/schema/task
http://www.springframework.org/schema/task/spring-task-4.0.xsd"
	default-lazy-init="false">
	<!-- 配置c3p0的连接池 -->
	<bean id="dataSource"
class="com.mchange.v2.c3p0.ComboPooledDataSource">
		<!-- 注入属性值 -->
		<property name="driverClass" value="com.mysql.cj.jdbc.Driver">
</property>
		<property name="jdbcUrl"
			value="jdbc:mysql://localhost:3306/hb1?serverTimezone=UTC">
</property>
		<property name="user" value="root"></property>
		<property name="password" value="AQL271422"></property>
	</bean>
```

```
	<!-- create object -->
	<bean id="orderDao" class="cn.java.ZhuanZhang.Dao.OrderDao">
		<!-- 注入jdbctemplate -->
		<property name="jdbcTemplate" ref="jdbcTemplate"></property>
	</bean>
	<bean id="orderService" class="cn.java.ZhuanZhang.Service.OrderService">
		<!-- 注入属性 -->
		<property name="orderDao" ref="orderDao"></property>
	</bean>
	<!-- 注入模板 -->
	<bean id="jdbcTemplate"
class="org.springframework.jdbc.core.JdbcTemplate">
	<!-- 把dataSource传递到模板对象里面 -->
	<property name="dataSource" ref="dataSource"></property>
	</bean>
</beans>
```

4，测试代码

```
package cn.java.testSet;
import org.junit.Test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import cn.java.ZhuanZhang.Service.OrderService;
public class TestZhuanZhang {
	@Test
	public void testTemplate(){
		ApplicationContext context =
				new ClassPathXmlApplicationContext("bean2.xml");
		OrderService book = (OrderService)context.getBean("orderService");
		book.accountMoney();
	}
}
```

5，运行结果

```
mysql> select *from user;
+----+-----------+--------+
| id | username  | salary |
+----+-----------+--------+
|  1 | anqili    |    900 |
|  2 | linmingjun |   1100 |
+----+-----------+--------+
2 rows in set (0.00 sec)
```