## 一,(事务管理一)中代码产生的问题

- 1,如果anqili少了10之后,出现异常,linmingjun的钱数是不会增加的,就相当于此时linmingjun的前无缘无故的少了100元。
- 2,解决办法
  - (1) 添加事务解决, 出现异常的时候进行事务的回滚。

## 二,声明式事务管理(xml文件配置)

- 1,配置文件方式使用
- (1) 配置事务管理器
- <!-- 第一步配置事务管理器 -->
  <bean id="transactionManager"
- - (2) 配置事务的增强
- <!-- 第二步 配置事务的增强 -->

<tx:advice id="txadvice" transaction-manager="transactionManager">

<!-- 做事务的操作 -->

<tx:attributes>

<!-- 设置进行事务操作的方法的匹配规则,name表示对account开头的方法进行事务操作 -->

<tx:method name="account\*" propagation="REQUIRED"/>
</tx:attributes>

</tx:advice>

- (3) 配置切面
- <!-- 第三步 配置切面 -->

<aop:config>

<!-- 配置 切入点-->

<aop:pointcut expression="execution(\*

cn.java.ZhuanZhang.Service.OrderService.\*(..))" id="pointCut"/>

<!-- 配置切面 -->

```
<aop:advisor advice-ref="txadvice" pointcut-ref="pointCut"/>
    </aop:config>
2, 具体的配置文件信息
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
    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"
xmlns:aop="http://www.springframework.org/schema/aop"
    xsi:schemaLocation="
    http://www.springframework.org/schema/beans
https://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd
    http://www.springframework.org/schema/aop
https://www.springframework.org/schema/aop/spring-aop.xsd
    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"</pre>
class="com.mchange.v2.c3p0.ComboPooledDataSource">
         <!-- 注入属性值 -->
         com.mysql.cj.jdbc.Driver">
</property>
         property name="jdbcUrl"
             value="jdbc:mysql://localhost:3306/hb1?serverTimezone=UTC">
</property>
         cont = "user" value = "root" >
```

```
</bean>
    <!-- 第一步配置事务管理器 -->
    <bean id="transactionManager"</pre>
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
       <!-- 注入DataSource -->
       content
    </bean>
    <!-- 第二步 配置事务的增强 -->
    <tx:advice id="txadvice" transaction-manager="transactionManager">
       <!-- 做事务的操作 -->
       <tx:attributes>
           <!-- 设置进行事务操作的方法的匹配规则,name表示对account开头的方法
进行事务操作 -->
           <tx:method name="account*" propagation="REQUIRED" />
       </tx:attributes>
    </tx:advice>
    <!-- 第三步 配置切面 -->
    <aop:config>
    <!-- 配置 切入点-->
       <aop:pointcut expression="execution(*
cn.java.ZhuanZhang.Service.OrderService.*(..))" id="pointCut"/>
       <!-- 配置切面 -->
       <aop:advisor advice-ref="txadvice" pointcut-ref="pointCut"/>
    </aop:config>
    <!-- create object -->
   <bean id="orderDao" class="cn.java.ZhuanZhang.Dao.OrderDao">
       <!-- 注入jdbctemplate -->
       cproperty name="jdbcTemplate" ref="jdbcTemplate" > /property>
    </bean>
    <bean id="orderService" class="cn.java.ZhuanZhang.Service.OrderService">
       <!-- 注入属性 -->
       corderDao" ref="orderDao">
    </bean>
    <!-- 注入模板 -->
    <bean id="jdbcTemplate"</pre>
class="org.springframework.jdbc.core.JdbcTemplate">
       <!-- 把dataSource传递到模板对象里面 -->
       content
    </bean>
</beans>
```