



链滴

test01

作者: [java233](#)

原文链接: <https://ld246.com/article/1475902910906>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

<pre class="brush: java;toolbar: false;auto-links: false">
//JDBC中使用事务来模拟转帐
public class TransactionDemo1 {

    // 模拟转账成功时的业务场景
    public void testTransaction1() {
        Connection conn = null;
        PreparedStatement ps = null;
        ResultSet rs = null;
        try {
            conn = JdbcUtils.getConnection();
            conn.setAutoCommit(false);// 通知数据库开启事务(start transaction)
            String sql1 = "update account set money=money-100 where name='A'";
            ps = conn.prepareStatement(sql1);
            ps.executeUpdate();
            String sql2 = "update account set money=money+100 where name='B'";
            ps = conn.prepareStatement(sql2);
            ps.executeUpdate();
            conn.commit();// 上面的两条SQL执行Update语句成功之后就通知数据库提交事务(commit)
            System.out.println("成功");
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            JdbcUtils.release(conn, ps, rs);
        }
    }

    // 模拟转账过程中出现异常导致有一部分SQL执行失败后让数据库自动回滚事务
    public void testTransaction2() {
        Connection conn = null;
        PreparedStatement ps = null;
        ResultSet rs = null;
        try {
            conn = JdbcUtils.getConnection();
            conn.setAutoCommit(false);
            String sql1 = "update account set money=money-100 where name='A'";
            ps = conn.prepareStatement(sql1);
            ps.executeUpdate();
            // 用这句代码模拟执行完SQL1之后程序出现了异常而导致后面的SQL无法正常执行,
            // 事务也无法正常提交, 此时数据库会自动执行回滚操作
            int x = 1 / 0;
            String sql2 = "update account set money=money+100 where name='B'";
            ps = conn.prepareStatement(sql2);
            ps.executeUpdate();
            conn.commit();
            System.out.println("成功");
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            JdbcUtils.release(conn, ps, rs);
        }
    }

    // 模拟转账过程中出现异常导致有一部分SQL执行失败时手动通知数据库回滚事务
    @Test
}

```

```
public void testTransaction3() {
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try {
        conn = JdbcUtils.getConnection();
        conn.setAutoCommit(false);
        String sql1 = "update account set money=money-100 where name='A'";
        ps = conn.prepareStatement(sql1);
        ps.executeUpdate();
        // 用这句代码模拟执行完SQL1之后程序出现了异常而导致后面的SQL无法正常执行, 事务也
        // 无法正常提交
        int x = 1 / 0;
        String sql2 = "update account set money=money+100 where name='B'";
        ps = conn.prepareStatement(sql2);
        ps.executeUpdate();
        conn.commit();
        System.out.println("成功");
    } catch (Exception e) {
        try {
            // 捕获到异常之后手动通知数据库执行回滚事务的操作
            conn.rollback();
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
        e.printStackTrace();
    } finally {
        JdbcUtils.release(conn, ps, rs);
    }
}
}
} </pre>
```