**测试用例清单**

**（一）AllInOneUnitTest 类（共 14 项）**

| **编号** | **测试方法** | **功能点** | **输入数据** | **预期结果** | **实际结果** | **结论** |
| --- | --- | --- | --- | --- | --- | --- |
| TC01 | testDatabaseConnection() | 测试 SQL Server 连接是否成功 | 正确数据库配置 | “SQL Server 数据库连接成功！” | 一致 | ✅ |
| TC02 | testInvalidOrderId() | 测试无效订单查询 | order\_id = DD99999 | 返回“未查询到任何用户” | 一致 | ✅ |
| TC03 | testValidOrderId() | 测试有效订单查询 | order\_id = DD00007 | 查询到用户及商品编号 | 一致 | ✅ |
| TC04 | testInsertTradeRecord() | 测试交易记录插入逻辑 | 订单号 DD00007 | Trade 表插入成功 | 一致 | ✅ |
| TC05 | testUpdateGoodStatus() | 测试商品状态修改 | work\_id = 1068 | 状态从 available → frozen | 一致 | ✅ |
| TC06 | testTradeIdGeneration() | 测试交易号生成算法 | 任意订单号 | trade\_id = TD + 时间戳 | 一致 | ✅ |
| TC07 | testSessionDataSave() | 测试购买人信息是否存入 session | 模拟 HttpSession | 存入1条购买人数据 | 一致 | ✅ |
| TC08 | testReservationInsert() | 测试预订信息写入 Reservation 表 | 订单号 DD00007 | 预订信息成功插入 | 一致 | ✅ |
| TC09 | testMultipleOrderQuery() | 测试多订单查询结果 | 多个 order\_id | 查询结果按数量返回 | 一致 | ✅ |
| TC10 | testUnavailableTradeQuery() | 测试冻结与可用状态过滤 | 含多状态数据 | 仅返回状态为 frozen/available | 一致 | ✅ |
| TC11 | testUIStyleAvailable() | 测试前端样式：可用状态颜色 | 商品状态 available | bg-[blue-100 text-blue]-800 | 一致 | ✅ |
| TC12 | testUIStyleFrozen() | 测试前端样式：冻结状态颜色 | 商品状态 frozen | bg-[gray-100 text-gray]-800 | 一致 | ✅ |
| TC13 | testErrorHandling() | 测试异常输入处理 | 空 order\_id | 捕获异常并提示 | 一致 | ✅ |
| TC14 | testCloseConnection() | 测试数据库关闭逻辑 | 无 | 输出“SQL Server 数据库连接已关闭！” | 一致 | ✅ |