

武汉大学校企合作实践项目 2026 - 模拟股票交易对敲撮合系统

Elvis Wang

2026-01-15

Contents

1. 引言	1
2. 项目目标	1
2.1. 基础目标	2
2.1.1. 交易转发	2
2.1.2. 对敲风控	2
2.1.3. 模拟撮合	2
2.2. 高级目标	2
2.2.1. 行情接入	2
2.2.2. 撤单支持	2
2.2.3. 性能优化	2
2.2.4. 管理界面	2
2.2.5. 数据分析	2
3. 数据结构	2
3.1. 交易订单	2
3.2. 交易撤单	3
3.3. 行情信息	3
3.4. 交易所订单确认回报	3
3.5. 交易所订单非法回报	3
3.6. 交易所订单成交回报	4
3.7. 交易所撤单确认回报	4
3.8. 交易所订单非法回报	5
4. 项目预计产出	5

1. 引言

A 股市场中，交易订单进入市场后由交易所主机进行处理，通过交易规则校验后会进入撮合流程。当前 A 股市场采用“价格优先、时间优先”的竞价撮合机制。

对于交易客户来说，交易所会分配一个股东号。为了避免内部虚假交易风险，交易所禁止同股东号下对敲交易。交易客户需要在交易订单送去交易所之前进行风控，过滤会触发对敲交易的订单。

对于证券行业从业人员而言，深入理解订单撮合规则和风控规则，是每一个人的基本要求。设计实现符合实际的撮合系统和（或）风控系统，是掌握交易核心逻辑、熟悉证券交易规则的关键实践途径。

本次实践项目以 A 股主板股票交易为模拟场景，要求参与方设计并实现可扩展的股票交易订单对敲撮合系统。通过实践，帮助参与方掌握交易风控和交易撮合的核心逻辑、熟悉证券交易的基础规则，提升逻辑建模、代码实现与系统测试能力，为后续参与交易系统相关工作积累基础。

2. 项目目标

本次实践项目的目标分为基础目标（必选）与高级目标（可选）：基础目标聚焦核心逻辑与基础功能实现，旨在考核对证券交易规则的理解、基础逻辑建模与代码实现能力，为必完成项；高级目标聚焦功能拓展与性能优化，旨在拔高标准，检验系统设计深化能力与问题解决进阶能力，为可选完成项。

2.1. 基础目标

2.1.1. 交易转发

1. 实现模拟交易转发逻辑，接受输入交易订单，输出交易订单
2. 实现模拟回报转发逻辑，接受输入回报，输出回报
3. 实现基本的交易校验功能，对于简单的无效订单，输出一条交易非法的回报

2.1.2. 对敲风控

1. 实现对于会触发对敲交易的交易订单的过滤，不输出该交易订单
2. 实现对于会触发对敲交易的交易订单，输出一条交易非法的回报

2.1.3. 模拟撮合

1. 实现对敲交易订端自动撮合，返回成交回报
2. 被匹配的对手方交易订单需要从交易所侧撤回
3. 设计成交价生成算法
4. 处理零股成交

2.2. 高级目标

2.2.1. 行情接入

1. 考虑行情信息：能读取解析输入的行情信息
2. 在撮合时考虑行情信息：撮合时需要保证买价、卖价和对手方价格保持一致

2.2.2. 撤单支持

1. 处理输入的撤单请求

2.2.3. 性能优化

1. 分析性能瓶颈
2. 面向吞吐量优化
3. 面向低延时优化

2.2.4. 管理界面

1. 实现可视化前端展示界面框架
2. 实现可视化展示当前订单状态
3. 实现手动输入交易订单入口

2.2.5. 数据分析

1. 实现交易历史记录存储功能，便利离线分析
2. 实现交易历史记录分析功能，输出分析结果

3. 数据结构

3.1. 交易订单

以 JSON Object 格式输入：

- clOrderId (char[16]): 订单的唯一编号
- market (char[4]): 订单交易的市场
 - XSHG: 上交所
 - XSHE: 深交所
 - BJSE: 北交所
- securityId (char[6]): 订单交易的股票代码 (如 "600030")
- side (char[1]): 订单买卖方向

- B: 买
- S: 卖
- qty (uint32): 订单数量
- price (double): 订单价格
- shareholderId (char[10]): 股东号

3.2. 交易撤单

以 JSON Object 格式输入:

- clOrderId (char[16]): 撤单的唯一编号
- origClOrderId (char[16]): 待撤订单的唯一编号
- market (char[4]): 待撤订单交易的市场
 - XSHG: 上交所
 - XSHE: 深交所
 - BJSE: 北交所
- securityId (char[6]): 待撤订单交易的股票代码 (如 "600030")
- shareholderId (char[10]): 待撤订单的股东号
- side (char[1]): 待撤订单买卖方向
 - B: 买
 - S: 卖

3.3. 行情信息

以 JSON Array 格式输入:

- market (char[4]): 行情对应的市场
 - XSHG: 上交所
 - XSHE: 深交所
 - BJSE: 北交所
- securityId (char[6]): 行情对应的股票代码 (如 "600030")
- bidPrice (double): 行情对应的最新买价
- askPrice (double): 行情对应的最新卖价

3.4. 交易所订单确认回报

以 JSON Object 格式输入:

- clOrderId (char[16]): 订单的唯一编号
- market (char[4]): 订单交易的市场
 - XSHG: 上交所
 - XSHE: 深交所
 - BJSE: 北交所
- securityId (char[6]): 订单交易的股票代码 (如 "600030")
- side (char[1]): 订单买卖方向
 - B: 买
 - S: 卖
- qty (uint32): 订单数量
- price (double): 订单价格
- shareholderId (char[10]): 股东号

3.5. 交易所订单非法回报

以 JSON Object 格式输入:

- clOrderId (char[16]): 订单的唯一编号
- market (char[4]): 订单交易的市场
 - XSHG: 上交所
 - XSHE: 深交所
 - BJSE: 北交所
- securityId (char[6]): 订单交易的股票代码 (如 "600030")
- side (char[1]): 订单买卖方向
 - B: 买
 - S: 卖
- qty (uint32): 订单数量
- price (double): 订单价格
- shareholderId (char[10]): 股东号
- rejectCode (int32): 非法的代码
- rejectText (char[64]): 非法的原因说明

3.6. 交易所订单成交回报

以 JSON Object 格式输入:

- clOrderId (char[16]): 订单的唯一编号
- market (char[4]): 订单交易的市场
 - XSHG: 上交所
 - XSHE: 深交所
 - BJSE: 北交所
- securityId (char[6]): 订单交易的股票代码 (如 "600030")
- side (char[1]): 订单买卖方向
 - B: 买
 - S: 卖
- qty (uint32): 订单数量
- price (double): 订单价格
- shareholderId (char[10]): 股东号
- execId (char[12]): 成交的唯一编号
- execQty (uint32): 成交数量
- execPrice (double): 成交价格

3.7. 交易所撤单确认回报

以 JSON Object 格式输入:

- clOrderId (char[16]): 撤单的唯一编号
- origClOrderId (char[16]): 已撤订单的唯一编号
- market (char[4]): 已撤订单交易的市场
 - XSHG: 上交所
 - XSHE: 深交所
 - BJSE: 北交所
- securityId (char[6]): 已撤订单交易的股票代码 (如 "600030")
- shareholderId (char[10]): 已撤订单的股东号
- side (char[1]): 已撤订单买卖方向
 - B: 买
 - S: 卖
- qty (uint32): 已撤订单数量
- price (double): 已撤订单数量

- cumQty (uint32): 已撤订单累计总成交数量
- canceledQty (uint32): 已撤订单已撤数量

3.8. 交易所订单非法回报

以 JSON Object 格式输入:

- clOrderId (char[16]): 撤单的唯一编号
- origClOrderId (char[16]): 待撤订单的唯一编号
- rejectCode (int32): 非法的代码
- rejectText (char[64]): 非法的原因说明

4. 项目预计产出

1. 项目设计文档
2. 实现了全部目标功能的代码
3. 可演示的程序运行环境
4. 答辩材料