**Тестовое задание C++ (NTPro)**

**📈 Легенда**

Ты стал новым руководителем отдела биржи. Поздравляем! 🎉

Но выяснилось, что в твоём отделе уже несколько десятков лет торговля ведется по старинке! Все измучены миллионами звонков от трейдеров, выставляющих заявки по телефону.

Ты хочешь всё это автоматизировать, чтобы клиенты могли торговать через компьютер. Тогда все могли бы сидеть в тишине и обрабатывать их заявки ☺️

К счастью, у тебя есть для этого всё — Linux, C++ и полный гитхаб различных библиотек.

**🔮 Что необходимо сделать**

Необходимо написать простую биржу с клиент-серверной архитектурой. Для упрощения задачи, биржа будет торговать только долларами (USD) за рубли (RUB).

Сервер должен принимать заявки на покупку или продажу долларов за рубли от разных клиентов по определённой цене.

В случае, если цена на покупку и цена на продажу у нескольких клиентов пересекается — нужно заключать сделки между ними. В этом случае купленный объём будет зачисляться на баланс клиентам.

**Условия:**

**Сервер**

* Поддерживает подключения нескольких клиентов одновременно.
* Даёт возможность просмотра клиентом всех своих активных торговых заявок, всех своих совершенных сделок и своего баланса.

**Клиент**

* Подключается к серверу и реализует все его возможности.

**Торговая логика**

* Торговая заявка содержит объём, цену и сторону (покупка/продажа).
* Торговая заявка активна до тех пор, пока по ней есть активный объём.
* Если две заявки пересекаются по цене — для сделки выбирается цена более ранней заявки.
* Если заявка пересекается с несколькими заявками по другой стороне — приоритет в исполнении отдаётся заявкам с максимальной ценой для заявок на покупку и с минимальной ценой для заявок на продажу.
* Возможно частичное исполнение заявки. (см. пример)
* Баланс клиента не ограничен — он может торговать в минус.

📝 **Пример:**

* Пользователь 1 выставил заявку на покупку **10** USD за RUB по цене 62.
* Пользователь 2 выставил заявку на покупку **20** USD за RUB по цене 63.
* Пользователь 3 выставил заявку на продажу **50** USD за RUB по цене 61.
* Сервер сматчил эти заявки между собой. Получилась две сделки:
  + На **20**$ по цене 63 между П2 и П3.

Важно! Мы начинаем исполнение именно с этой заявки, т.к. цена в П2 больше, чем цена в П1

* + На **10**$ по цене 62 между П1 и П3.
* Теперь на балансе пользователей:
  + Пользователь 1: **10** USD, **-620** RUB.
  + Пользователь 2: **20** USD, **-1260** RUB.
  + Пользователь 3: **-30** USD, **1880** RUB.
* Торговая заявка пользователя 3 (частично исполненная на **30$**) осталась активной на **20$**.

🎈 **Бонусы! Опционально реализуйте:**

* Возможность просмотра клиентом котировок 📈
* Получение отчёта о совершённой сделке в момент её заключения всеми сторонами-участниками.
* Возможность отмены активных заявок клиентами. 🚫
* Сохранение истории всех заявок и сделок в базу данных (например, PostgreSQL). 💽
* Авторизацию клиента с паролем. 🔑
* GUI к бирже на QT. 🤯

**💻 Требования по реализации**

* Клиент может представлять собой обычное консольное приложение.
* Все основные функции должны быть покрыты тестами.

**🔨 Технические требования**

* Проект должен быть написан с использованием C++17/20.
* Разрешается использование сторонних библиотек.
* Проект должен собираться с использованием CMake.
* Описание, инструкция по сборке и инструкция по использованию проекта должны быть в файле [**README.md**](http://README.md)
* Проект должен быть выполнен в git-репозитории, ссылку на который необходимо предоставить.

Для упрощения задачи мы [**приготовили каркас**](https://bitbucket.org/ntprog/templateofclientservermatching/src/master/) для нашего тестового задания. Он призван сэкономить ваше время, а не навязать архитектуру реализации. Его можно не использовать, если вы не хотите.