ООО «Дольмен»

ПРОГРАММА ДЛЯ ЭВМ
«LiteBot»

Документация, содержащая описание
функциональных характеристик
экземпляра программного обеспечения

Версия 0.2.0

Санкт-Петербург, 2026

Содержание

1. Предназначение ПО.....3
2. Описание функциональных характеристик.....3
3. Программные и аппаратные требования.....5
4. Состав системы.....6
5. Эксплуатация системы.....8

1. Предназначение ПО

LiteBot — программа для автоматизированной торговли на Московской бирже (ПАО Московская Биржа, далее — биржа). Программа получает рыночные данные биржи, исполняет торговые алгоритмы (стратегии), выставляет и сопровождает заявки, ведёт учёт позиций и финансового результата, применяет риск-ограничения и предоставляет оператору средства управления и контроля.

Программа предназначена для профессиональных участников рынка ценных бумаг, их клиентов и иных лиц, допущенных к торгам на бирже и подключающих торговые алгоритмы к её программно-техническим средствам.

Программа решает задачу автоматического исполнения торговых алгоритмов на биржевом рынке с контролем рисков: от приёма рыночных данных до выставления заявок и учёта результата, без участия человека в каждой операции и под контролем оператора.

Область применения:

- фондовый и валютный рынки биржи;
- срочный рынок биржи: фьючерсы и опционы;
- алгоритмическая торговля, исполнение заявок, поддержание котировок, ручная торговля оператора через программный интерфейс.

2. Описание функциональных характеристик

1. Приём рыночных данных. Программа принимает рыночные данные биржи по протоколу SIMBA (биржевой протокол многоадресной рассылки) для всех поддерживаемых рынков, в том числе из основного и резервного каналов. Она получает лучшие цены, сделки, торговые статусы инструментов и торговых сессий и восстанавливает состояние после пропуска данных и при запуске в течение торгового дня. Как дополнительный источник для стратегий могут подключаться данные внешних площадок (функция необязательная, по умолчанию выключена).

2. Книги заявок. Программа строит книги заявок по инструментам, включая полную книгу с очерёдностью заявок на ценовом уровне. Стратегиям доступны ценовые уровни с объёмом и числом заявок и очередь перед

собственной заявкой; оператор просматривает книгу заявок инструмента через интерфейс управления.

3. Справочник инструментов. В течение торгового дня программа ведёт справочник инструментов биржи: шаг цены, лот, точность, стоимость шага, комиссии, ценовые ограничения. Торгуемый контракт срочного рынка выбирается автоматически по заданному правилу (например, ближайший квартальный).

4. Выставление и сопровождение заявок. Программа выставляет заявки на биржу по протоколу TWIME (биржевой протокол приёма заявок) на всех поддерживаемых рынках. Типы заявок: лимитная на день, лимитная «исполнить немедленно или отклонить», лимитная только пассивная, рыночная. Поддерживаются изменение цены и объёма, отмена заявки, отмена всех заявок по инструменту, массовая отмена по счёту. До отправки объёмы стратегии переводятся в лоты биржи, а параметры заявки проверяются по справочнику инструментов.

5. Сессии с биржевыми шлюзами. Программа устанавливает и поддерживает сессии со шлюзами приёма заявок, восстанавливает их после разрыва с получением пропущенных сообщений и использует резервные адреса шлюза.

6. Исполнение торговых стратегий. Программа исполняет торговые стратегии, подключаемые через программный интерфейс стратегий. Одновременно на одном счёте может работать любое число портфелей (стратегия с её инструментами, параметрами, позициями и ограничениями). Портфель можно запустить, остановить, приостановить и возобновить, параметры стратегии — изменить во время торгов. Стратегия, проверенная на исторических данных в системе исторического тестирования правообладателя, исполняется в торговле без изменений. В комплект поставки входит демонстрационная стратегия «лестница заявок».

7. Расчёт признаков стратегий. Признаки стратегии (величины, вычисляемые из рыночных данных) задаются выражениями в конфигурации. Набор признаков компилируется в один граф зависимостей; при событии пересчитываются только признаки, зависящие от него.

8. Комплект разработчика. Комплект разработчика (SDK) содержит программу в виде статической библиотеки, программный интерфейс

стратегий в исходном тексте и учебные примеры. Пользователь собирает исполняемый файл с собственными стратегиями; у него та же командная строка, конфигурация, интерфейс управления и журналы, он выводит версию программы и сборку пользователя.

9. Учёт позиций и финансового результата. Программа ведёт позиции по каждому портфелю и по счёту в целом, сохраняет их и восстанавливает при перезапуске. Рассчитываются средняя цена, реализованный и нереализованный результат, комиссии, оборот, экспозиция по портфелям и инструментам.

10. Риск-контроль. До отправки на биржу каждая заявка проверяется: цена, объём, стоимость, частота заявок, отклонение цены от рынка, предельная позиция с учётом рабочих заявок, экспозиция, предотвращение сделок с самим собой. Ограничения задаются на счёт в целом, общие для всех портфелей и для отдельного портфеля. Ограничение убытка и просадки (стоп-лосс) срабатывает с выбранной реакцией: запрет новых заявок, только сокращение позиции, закрытие позиции; сработавшее ограничение действует, пока его не снимет оператор. Все заявки можно аварийно снять одной командой.

11. Поведение при сбоях. Программа снимает заявки при запуске, после восстановления связи и при остановке. Торговля приостанавливается, если нет свежих рыночных данных, торги инструментом приостановлены или справочник биржи расходится с ожидаемыми параметрами инструмента, и возобновляется после устранения причины.

12. Управление. Программный интерфейс управления работает по протоколу HTTP с авторизацией по токену и отдельными правами на операции, снимающие защитные ограничения. Через него выполняются управление портфелями, ручная торговля оператора, управление риск-ограничениями и просмотр состояния: заявки, позиции, результат, сессии, инструменты. Для систем мониторинга предусмотрены проверки работоспособности и готовности.

13. Журналирование и мониторинг. Программа ведёт журнал всех запросов на биржу и всех ответов биржи с отметками времени и журнал событий в машиночитаемом формате (строки JSON и (или) колоночный формат Parquet). Журналы торгового дня переводятся в формат Parquet, формируется отчёт о времени обработки за день по стадиям, часам и

инструментам. Метрики работы и торговли выдаются в формате Prometheus; возможна выгрузка событий и журналов в Apache Kafka.

14. Режимы проверки и сертификации. Программа работает в демонстрационном режиме (синтетический рынок с имитатором шлюза биржи, без подключения к бирже), проигрывает записанный рыночный трафик, содержит встроенный имитатор биржевого шлюза и режим исполнения заявок в памяти, проверяет конфигурацию без запуска торговли и выполняет сценарии сертификации программно-технических средств участника, предусмотренные порядком сертификации биржи.

Ограничения программы:

- программа не отправляет заявки на площадки, отличные от Московской биржи;
- позиции ведутся по данным программы и с отчётами брокера и клиринга автоматически не сверяются;
- подключение к боевым шлюзам биржи требует прохождения сертификации программно-технических средств в порядке, установленном биржей.

3. Программные и аппаратные требования

Программные требования:

- Операционная система: Linux с systemd, архитектуры x86_64 и aarch64. Программа проверена на Astra Linux 1.8, ALT Linux p11, RHEL 9; поддерживаются RHEL 9 и совместимые (Rocky Linux, AlmaLinux, Oracle Linux) и российские дистрибутивы Linux того же поколения.
- Исполняемый файл статически скомпонован и не зависит от системных библиотек. СУБД, веб-сервер, сервер приложений и среда исполнения (виртуальная машина языка) не требуются.
- Синхронизация времени: chrony или NTP; для измерения задержек — PTP.
- Для автоматизированной установки на рабочем месте администратора: ansible-core 2.14 или новее, коллекция `ansible.posix`, доступ по SSH к серверу с правами sudo.

– Доступы к бирже: подключение к биржевым сервисам рыночных данных и приёма заявок, логины и пароли шлюзов, торговый счёт и код клиента — предоставляются биржей или брокером.
– Внешние системы по выбору пользователя: Prometheus, Apache Kafka.

Аппаратные требования:

– Процессор: архитектура x86_64 или aarch64, не менее 4 ядер; для торговли рекомендуется выделить одно ядро торговому потоку.
– Оперативная память: не менее 8 ГБ; разрешение на блокировку страниц в памяти.
– Хранилище данных: не менее 50 ГБ под журналы (объём зависит от срока хранения).
– Сеть: подключение к программно-техническим средствам биржи — приём многоадресной рассылки рыночных данных и TCP-соединения со шлюзами приёма заявок.

4. Состав системы

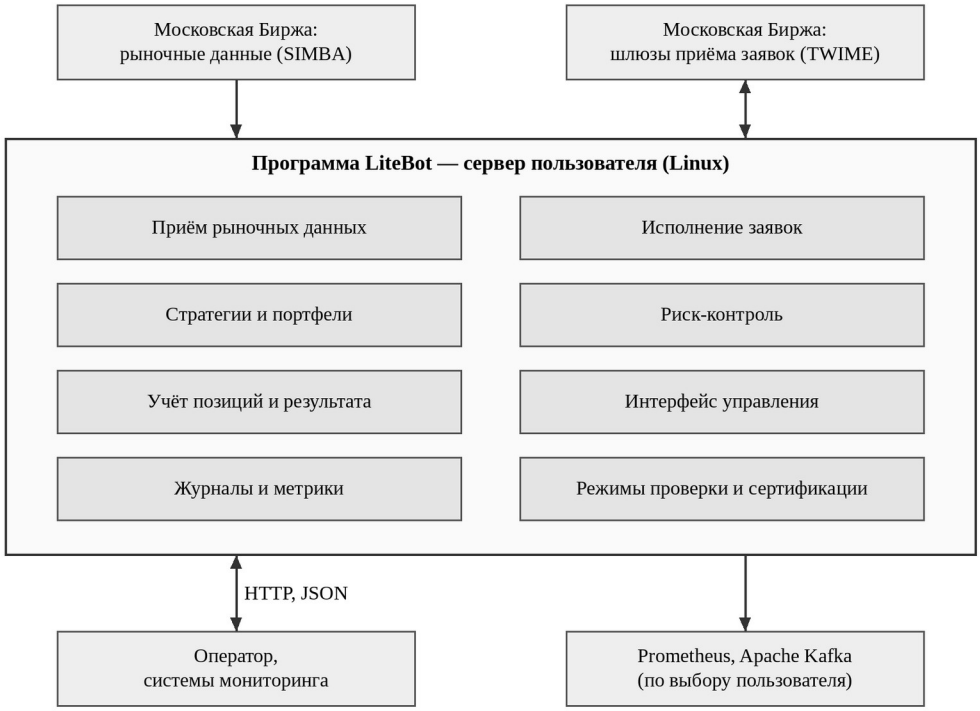
Описание технической инфраструктуры

Язык программирования — Rust.

Используемая платформа — не требуется: программа поставляется как один исполняемый файл для Linux, работает как служба systemd, конфигурация задаётся файлом формата TOML.

Используемые сторонние компоненты — библиотеки с открытым исходным кодом под разрешительными лицензиями (MIT, BSD, Apache 2.0, Zlib, Unlicense), включённые в исполняемый файл статически: библиотеки на языке Rust (чтение конфигурации и формат JSON, командная строка, HTTP-сервер интерфейса управления, сетевые сокеты, формат Parquet, выгрузка в Apache Kafka) и библиотеки на языке C (librdkafka, OpenSSL, zlib, Zstandard). Компоненты под лицензиями GPL, LGPL, AGPL, MPL не используются. Компоненты не обращаются к серверам их разработчиков и не содержат механизмов удалённого управления, обновления или лицензионных ключей. Полный перечень с версиями и тексты лицензий — файл `THIRD-PARTY-NOTICES.md` в комплекте поставки.

Общая структура системы



Приём рыночных данных — получение данных биржи по протоколу SIMBA, построение книг заявок, ведение справочника инструментов.

Стратегии и портфели — исполнение торговых алгоритмов, подключаемых через программный интерфейс стратегий; портфели со своими инструментами, параметрами и ограничениями.

Риск-контроль — проверка каждой заявки до отправки на биржу, ограничения счёта и портфелей, стоп-лоссы.

Исполнение заявок — сессии со шлюзами биржи по протоколу TWIME, выставление, изменение и снятие заявок.

Учёт позиций и результата — позиции по портфелям и счёту, их сохранение и восстановление, расчёт результата.

Интерфейс управления — программный интерфейс по протоколу HTTP (JSON) для оператора и систем мониторинга.

Журналы и метрики — журналы обмена с биржей и событий программы, метрики Prometheus, необязательная выгрузка в Apache Kafka.

Режимы проверки и сертификации — демонстрационный режим, проигрывание записей рынка, имитатор шлюза биржи, сценарии сертификации.

Комплект поставки: исполняемый файл `litebot`, имитатор шлюза `twime-mock`, схемы протоколов биржи, примеры конфигурации, сценарии установки Ansible, средство замера времени обработки, файлы `VERSION`, `LICENSE`, `THIRD-PARTY-NOTICES.md`. Комплект разработчика поставляется отдельно.

5. Эксплуатация системы

5.1. Запуск системы

Перед запуском убедитесь, что сервер соответствует требованиям раздела 3, а конфигурация заполнена по документам подключения биржи. Проверка конфигурации без подключения к бирже — `litebot check --config <файл>`. Запуск — `systemctl start litebot`. При запуске программа печатает версию, проверяет сервер и конфигурацию, восстанавливает позиции, устанавливает сессии со шлюзами, снимает заявки счёта, оставшиеся с прошлого запуска, и запускает портфели, когда получен справочник по их инструментам. Готовность к торговле подтверждает запрос `GET /health/ready` (код 200). Порядок установки и запуска приведён в руководстве по установке и эксплуатации.

5.2. Управление

Оператор управляет программой через программный интерфейс управления: запуск, остановка и приостановка портфелей, изменение параметров стратегий, ручные заявки, управление риск-ограничениями, аварийное снятие заявок. По умолчанию интерфейс доступен только с локального адреса сервера; удалённый доступ — через SSH-туннель или порт, открытый для доверенной сети. Состояние программы контролируется по запросам состояния, метрикам Prometheus и журналам. Рекомендуемые оповещения системы мониторинга: не проходит проверка работоспособности, растёт число отказов по токену, разорвана сессия шлюза в торговое время, не поступают рыночные данные, сработал стоп-лосс или запрет торговли, есть потери записей журнала.

5.3. Резервное копирование и восстановление

Программа не использует базу данных. Для восстановления работы достаточно сохранять:

– файл конфигурации (`/etc/litebot/litebot.toml`) и хранилище паролей и токенов;
– файл позиций портфелей (`positions.json` в каталоге данных программы);
– журналы заявок и событий — в течение срока хранения, установленного внутренними правилами пользователя и требованиями регулятора.

При обновлении предыдущая версия исполняемого файла сохраняется автоматически; возврат к ней выполняется сценарием отката (`ansible-playbook rollback.yml`). Позиции портфелей восстанавливаются при каждом запуске из файла позиций.

5.4. Обновление

Обновления не устанавливаются автоматически: пользователь сам решает, когда устанавливать новую версию. Программа не обращается к серверам правообладателя и не может управляться ими. Обновление выполняется сценарием `ansible-playbook deploy.yml`: новый исполняемый файл проверяется на установленной конфигурации, предыдущий сохраняется, служба перезапускается. Обновление рекомендуется выполнять вне торговой сессии. К каждому выпуску прилагается перечень изменений.

5.5. Техническая поддержка

Обращения принимает служба поддержки правообладателя — ООО «Дольмен»: support@dolmenfintech.ru, время работы: рабочие дни 09:00–18:00 МСК. Язык поддержки — русский. Поддержка включает консультации по установке и эксплуатации, разбор инцидентов по журналам, помощь в сертификации на бирже, выпуск исправлений. Обращение содержит версию программы (`litebot --version`), описание ситуации, время и, по возможности, фрагменты журналов. Журналы не содержат паролей и токенов.

Актуальная версия настоящего документа и другая документация программы размещены на сайте правообладателя: dolmenfintech.ru.

5.6. Завершение работы

Остановка — `systemctl stop litebot`, сигнал SIGTERM или запрос `POST /shutdown`. Портфели снимают свои заявки, шлюзы выполняют страховочную массовую отмену и закрывают сессии, журналы и позиции сохраняются. Время остановки ограничено; повторный сигнал завершает её

немедленно. Позиции при остановке не закрываются и восстанавливаются при следующем запуске.