empenoso 1 час назад Программирование на python

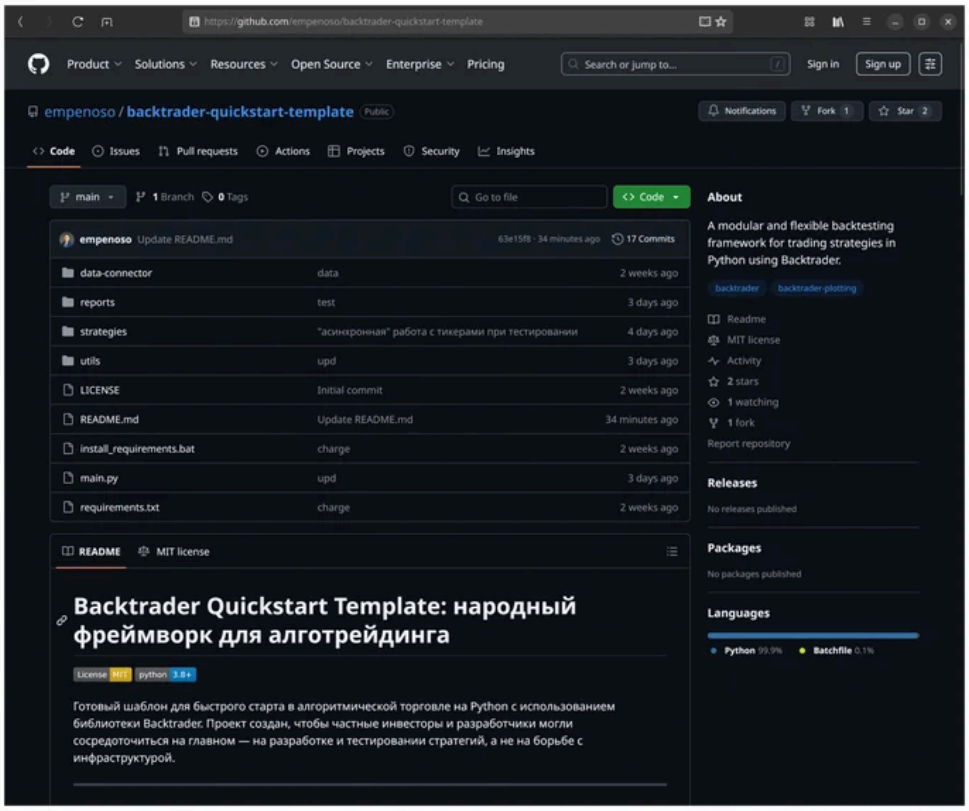
# Алготрейдинг на коленке: как сделать свой TradingView на Python

В мире алгоритмической торговли доминируют крупные фонды с их колоссальными ресурсами. Но что, если мы, частные инвесторы и разработчики, можем создать собственный мощный и доступный инструмент? Что, если больше не придётся зависеть от проприетарных платформ или писать с нуля сложную инфраструктуру для тестирования каждой новой идеи?

Сегодня у нас есть Python и такие мощные библиотеки, как Backtrader. Однако голый фреймворк — это лишь половина дела. Чтобы он стал по-настоящему народным инструментом, ему нужна удобная обвязка: готовая структура проекта, автоматический импорт стратегий, наглядные отчёты, тепловые карты для оптимизации и бесшовное подключение к API брокеров — не только российских, но надо начать с Мосбиржи.

Мы стремимся сделать инструмент таким же удобным, как TradingView. Простота в использовании и доступность всех функций для пользователей без глубокой технической экспертизы — мне кажется вот идеал. Чтобы каждый, кто заинтересован в алгоритмической торговле, мог без усилий внедрить свою стратегию, протестировать её и получить результаты, не проводя часы и дни за настройкой системы.

Эта статья — не просто описание проекта, а призыв к действию. Я предлагаю объединить усилия и **создать открытый стандарт для алготрейдинга на базе open source Backtrader, заточенный под реалии российского рынка**.




https://github.com/empenoso/backtrader-quickstart-template

Главная задача — построить открытый шаблон, который позволит частному инвестору, даже с небольшими навыками в программировании, сосредоточиться на главном — на разработке и тестировании стратегий, а не на борьбе с инфраструктурой. Мы создаём систему для марафона, а не для спринта: для классических стратегий с горизонтом в часы и дни, которая будет работать автономно и не требовать ежедневного внимания.

### Почему Backtrader, а не другая библиотека?

Выбор в пользу **Backtrader** был сделан не случайно. Прежде чем остановиться на этой библиотеке изучил несколько популярных open-source решений, каждое из которых имеет свои сильные стороны, но не соответствовало главной цели — создать простой и гибкий инструмент, который будет удобен для частных инвесторов на Московской бирже.

С одной стороны, существуют комплексные платформы, такие как OsEngine. Это мощное решение на C#, предлагающее готовый функционал «из коробки»: от графического интерфейса до подключения к различным брокерам. OsEngine представляет собой готовую экосистему, в которую интегрируются торговые роботы. Для тех, кто ищет законченный продукт и готов работать в его рамках, это отличный выбор. Однако для нашей цели — создания гибкого и легко кастомизируемого шаблона на Python — его архитектура оказалась менее подходящей, так как мы стремимся к максимальной простоте и модульности, свойственной конструкторам.

На другом полюсе — узкоспециализированные инструменты. **Freqtrade** — необычайно популярен в мире криптотрейдинга. Он обладает огромным сообществом и богатым функционалом, но весь этот функционал «заточен» под

цифровые активы. Для фондового рынка и тем более для Московской биржи потребуются серьёзные доработки.

Есть и проекты, выросшие из легендарного **Zipline** от Quantopian, например, **Ziplime**. Сам движок Zipline по-прежнему силён в бэктестинге, но после закрытия Quantopian все стало сложно.

Отдельно стоит упомянуть **Nautilus Trader** — это современный высокопроизводительный фреймворк, который часто называют «тяжёлой артиллерией» для HFT. Он невероятно быстр, поддерживает асинхронную архитектуру и ориентирован на работу с миллионами тиков в секунду. Но для большинства частных инвесторов такой уровень скорее избыточен. Для долгосрочных стратегий с горизонтом в часы и дни он выглядит как перегруженный инструмент.

На фоне этих решений **Backtrader** выделяется своей философией. Это не готовая платформа с жёсткими рамками, а библиотека-конструктор. Она даёт базовые «кубики», из которых можно собрать именно ту систему, которая нужна.

У неё есть три решающих преимущества: большое сообщество, качественная документация (и даже на русском, неофициальная, сделанная одним из пользователей) и проверенные интеграции с API российских брокеров.

Именно поэтому **Backtrader** кажется идеальной основой для «народного» шаблона: лёгкого, гибкого и понятного даже тем, кто делает первые шаги в алгоритмической торговле.

## Open source

Все компоненты нашего фреймворка — библиотека Backtrader и предлагаемый шаблон — распространяются под лицензией Open Source. Это означает не просто бесплатный доступ к коду, но и возможность совместно развивать и улучшать проект.

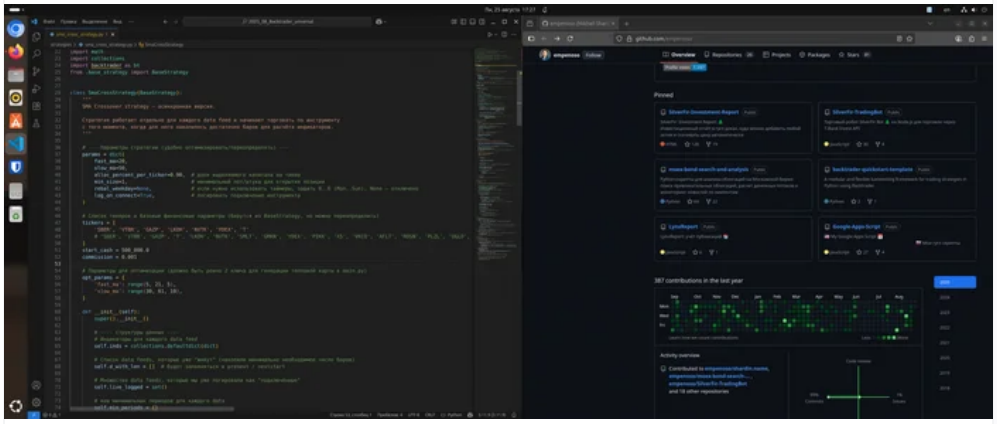
В мире open source качественные инструменты создаются коллективными усилиями и становятся доступны каждому, устраняя барьеры проприетарных решений. Наш «народный шаблон Backtrader» — часть движения, которое делает мощные инструменты алгоритмической торговли доступными без значительных финансовых затрат.

## Архитектура нашего шаблона: что «под капотом»?

В основе шаблона лежат три принципа: модульность, наглядность результатов и готовность к реальной торговле. Давайте заглянем «под капот» и посмотрим, как это устроено.

**1. Максимальная модульность: одна стратегия — один файл.**
Избавляемся от сложной структуры и запутанных зависимостей. Каждая ваша торговая идея — это отдельный Python-файл в папке strategies. Внутри этого файла вы описываете всё, что касается именно этой стратегии: логику входа и выхода, список тикеров для теста, размер комиссии брокера, начальный капитал и, что

самое важное, параметры для будущей оптимизации. Больше не нужно искать настройки по всему проекту. Система автоматически подхватывает все стратегии из этой папки, а в главном файле вы просто указываете, какую из них хотите запустить сегодня.



**2. Два режима работы: «тест» и «оптимизация».**

В основном файле проекта заложен простой переключатель. Хотите быстро проверить гипотезу на исторических данных? Включаете режим тестирования. Нужно подобрать лучшие параметры для вашей стратегии? Переключаетесь в режим оптимизации. Это позволяет сосредоточиться на задаче, не меняя код самой стратегии. Данные для тестов хранятся в отдельной папке data, а в настройках запуска вы лишь указываете нужный временной интервал, например, с 01.01.2022 по 31.12.2024.

**3. Информативные отчёты и визуализация.**

Поэтому по итогам каждого бэктеста фреймворк генерирует подробный текстовый отчёт в папку reports. В нём собраны ключевые метрики: чистая прибыль, годовая

доходность в сравнении с «купил и держи», максимальная просадка, профит-фактор и коэффициент Сортино.

```
--- ОТЧЕТ ПО БЭКТЕСТУ СТРАТЕГИИ: SmaCrossStrategy ---
Бумаги в тесте: ['SBER', 'VTBR', 'GAZP', 'LKOH', 'NVTK', 'YDEX', 'T']
Параметры: {'fast_ma': 20, 'slow_ma': 50, 'alloc_percent_per_ticker': 0.9, 'min_size':
1, 'rebal_weekday': None, 'log_on_connect': True}
Период тестирования: с 01.01.2018 по 12.08.2025
--- РЕЗУЛЬТАТЫ ---
Итоговая прибыль/убыток: 252 713.12 [50.54%]
Доходность (годовых): 5.44%
Результат 'Купил и держал': 205 648.56 [41.13%]
Максимальная просадка: 67 124.98 [13.42%]
Всего сделок: 128
Процент прибыльных сделок: 41.41% (53 из 128)
Фактор прибыли: 2.13
Коэффициент Сортино: 1.08
----------------------------------------------
```

Для режима оптимизации сознательно ограничились двумя параметрами. Почему? Это позволяет визуализировать результаты в виде тепловой карты. Глядя на неё, вы сразу видите, какие комбинации параметров дают наибольший профит-фактор, и можете выявить зоны переоптимизации, где стратегия становится неустойчивой.

**4. Бесшовный переход к автоторговле.**
Тестирование — это лишь первый шаг. Шаблон изначально спроектирован с возможностью лёгкого подключения к реальному рынку. В него заложена основа для интеграции с API.

В России есть несколько брокеров с открытыми API:

Брокер / Документация

Тинькофф (T-Invest) / https://developer.tbank.ru/invest/intro/intro

Алор / https://alor.dev/docs/

Финам / https://tradeapi.finam.ru/

Для Backtrader есть интеграции с API российских брокеров.

Когда ваша стратегия покажет стабильные результаты на истории, вы сможете активировать режим реальной торговли, просто изменив одну настройку.
Это превращает наш шаблон из простой «песочницы» в полноценный боевой инструмент.

**Как вы можете помочь?**

Я сделал первую рабочую версию шаблона, которую можно найти на GitHub. Это основа, скелет будущего шаблона, но чтобы он оброс «мясом» и стал по-настоящему удобным и «народным», ему нужна помощь сообщества. И здесь я обращаюсь к вам.

Одна из первых трудностей, с которой я столкнулся, — это визуализация результатов. Стандартная функция cerebro.plot() отлично справляется с одной ценной бумагой, но при тестировании портфеля из десятка акций график превращается в нечитаемую кашу, где сигналы и сделки сливаются в сплошную линию. Приблизить и рассмотреть детали невозможно. В идеале, нужно научить систему сохранять графики для каждого тикера в отдельный файл с высоким разрешением, чтобы можно было спокойно и детально изучать каждую сделку.

Мне удалось сохранить график в таком читаемом виде только за счёт махинаций с поворотом экрана на 90° через программу управления монитором.

Сделки только на первом графике показаны

Но технические задачи — это лишь часть дела. **Я буду невероятно благодарен за любую конструктивную критику. Возможно, вы видите архитектуру иначе?**

**Знаете, как сделать отчёты ещё информативнее или добавить новые метрики? Можете предложить более изящный способ управления стратегиями? Любые идеи приветствуются.**

Да, я знаю о существовании такого популярного комплекса, как QUIK. Я уважаю его возможности, но сознательно выбрал путь Python. Гибкость, открытость и возможность запустить торгового робота даже на крошечном Raspberry Pi, который будет автономно работать месяцами — вот те преимущества, которые, на мой взгляд, перевешивают.

Если вы разделяете идею создания «народного Backtrader» — подключайтесь. Даже небольшие улучшения или просто замечания помогут довести этот проект до уровня, когда частные инвесторы смогут использовать его так же легко, как крупные игроки используют свои проприетарные системы.

## Заключение

Главная цель этого проекта — создать не просто очередную библиотеку или набор скриптов, а универсальный инструмент для алгоритмической торговли, который будет доступен каждому. Мы строим открытый шаблон на Python, чтобы частные инвесторы могли сосредоточиться на разработке стратегий и анализе идей, а не на бесконечной настройке инфраструктуры. Народный Backtrader — это шаг к формированию сообщества, где знания и опыт становятся общим ресурсом. Вместе мы можем превратить этот проект в удобный, гибкий и по-настоящему народный стандарт для алгоритмической торговли на Московской бирже.

**Автор:** Михаил Шардин
Моя онлайн-визитка
Telegram «Умный Дом Инвестора»

26 августа 2025

**Программирование на python**
878 постов • 11.9K подписчика

Добавить пост | Подписаться | ...

**Правила сообщества**

Публиковать могут пользователи с любым рейтингом. Однако!

**Приветствуется:**...

Подробнее

Все комментарии | Автора

Раскрыть 2 комментария

Чтобы оставить комментарий, необходимо зарегистрироваться или войти