

[Горячее](#) [Лучшее](#) [Свежее](#) ...[Войти](#)

empenoso 2 месяца назад



Программирование на python



Запилил тут скрипт на Python: ищем самые 'живые' акции на Мосбирже, чтобы торговый робот не покупал воздух. Делюсь!

Если вы задумывались о системной торговле, то, скорее всего, уже слышали о Python библиотеке **Backtrader**. Это гибкий фреймворк для тестирования торговых стратегий на исторических данных, который к тому же может быть подключён к автоторговле через API российского брокера. В нём можно реализовать практически любую логику, от простого пересечения скользящих средних до сложных многофакторных моделей.

Я посадил торгового робота в стену. Он теперь за меня торгует

Однако даже самая изощрённая стратегия ничего не стоит, если протестирована на неликвидных бумагах - там, где в реальной торговле вы бы просто не смогли купить или продать по нужной цене. Именно поэтому работа с ликвидными акциями - ключ к достоверному тесту.

Ликвидность - это не про «красиво на графике», а про то, как на самом деле исполняются сделки, насколько проскальзывает цена и как часто ваши заявки останутся без исполнения. Здесь нам поможет Игорь Чечет - автор библиотек **AlorPy**, **TinkoffPy** и **FinamPy**, размещенных на GitHub, которые дают удобный способ подключиться к API этих трёх брокеров из Python. Эти инструменты и библиотека-обертка - фактически мост между Backtrader и живым рынком.

В статье будем скачивать исторические данные настолько глубоко, насколько это возможно и находить самые активно торгуемые акции по кварталам за последние 20 лет при помощи моего Python скрипта.

Ваша критика или поддержка идей, приведённых в статье приветствуется.

Обзор библиотек Игоря Чечета: AlorPy, TinkoffPy, FinamPy

Каждая из этих библиотек предназначена для взаимодействия с API соответствующего брокера - Т-Инвестиции, Алор и Финам. Они позволяют подключаться к торговому счёту, получать историю торгов, котировки в реальном времени, стакан заявок, выставлять и снимать ордера. Это делает их отличной

Войти

[Войти](#)[Создать аккаунт](#)[Забыли пароль?](#)

или продолжите с

[Войти с Яндекс ID](#)[Войти через VK ID](#)[Промокоды](#)[Работа](#)[Курсы](#)[Реклама](#)[Игры](#)[Пополнение Steam](#)

основой как для обучения алгоритмической торговли, так и для построения полноценных торговых роботов.

Почему библиотеки не публикуются через PyPI

Если вы попытаетесь выполнить `pip install TinkoffPy`, ничего не произойдёт. Эти библиотеки не размещены в PyPI - и на это есть уважительная причина. Сам автор объясняет так:

Игорь Чечет:

"Сам давно хочу сделать свои библиотеки в виде python packages для pip. Проблема в том, что я не могу синхронизироваться с версиями API брокера. Например, сделал сегодня сборку. Всё работает. А завтра, раз, и брокер меняет API. Да так, что моя библиотека перестает работать. Приходится править свою библиотеку. Т.к. мои библиотеки очень сильно зависят от API брокеров, то или брокеры ведут версии своих API и change log, или буду править по горячим следам."

По сути, нестабильность API со стороны брокеров делает публикацию стабильной версии библиотеки невозможной - приходится «патчить на лету».

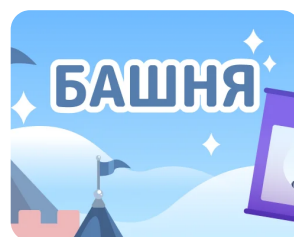
Как установить библиотеку в режиме разработки

Нет рекомендованного автором способа установки, но я выбрал использовать установку через режим разработки. Это позволяет подключить библиотеку в виде живого кода, и при необходимости оперативно вносить правки без повторной установки.



Пикабу Игры

+1000 бесплатных
онлайн игр



Башня

Аркады, Строительство, На
ловкость

Играть

Топ прошлой недели



AlexKud

51 пост



StenNews

empenoso / SilverFir-TradingBot_backtesting

<> Code Issues Pull requests Actions Projects Wiki Security

main SilverFir-TradingBot_backtesting / Backtrader_universal / data-conr

empenoso <https://habr.com/ru/articles/912528/>

Name
..
AlorPy
Data/Tinkoff
FinamPy
ISS-MOEX-client-master
TinkoffPy
!.txt
liquid_stocks_by_quarter.csv
liquid_stocks_by_quarter.py
setup.py
stock ticker search.py

Должна получится вот такая структура как на Гитхабе: https://github.com/empenoso/SilverFir-TradingBot_backtesting/tree/main/Backtrader_universal/data-connector

Алгоритм следующий:

1. Скачиваете его репозиторий с GitHub (на примере с Т-Инвестициями).
2. В корневую папку добавляете файл **setup.py**:

```
from setuptools import setup, find_packages

setup(
    name="TinkoffPy",
    version="1.0.0",
    description="Tinkoff Invest API Python wrapper",
    packages=find_packages(),
    include_package_data=True,
    python_requires=">=3.6",
)
```

Выполняете установку:

2 поста
Oskanov
8 постов

[Посмотреть весь топ](#)

Лучшие посты
недели

Рассылка Пикабу:
отправляем самые
рейтинговые материалы за 7
дней 🔥

Укажите [Подписаться](#)

Нажимая кнопку
«Подписаться на рассылку»,
я соглашаюсь с [Правилами](#)
[Пикабу](#) и даю согласие на
обработку персональных
данных.

Помощь	Правила
Кодекс Пикабу	соцсети
Команда	О
Пикабу	рекомендация
Моб.	х
приложение	О компании

Промокоды Биг Гик
Промокоды Lamoda
Промокоды МВидео
Промокоды Яндекс Директ
Промокоды Отелло
Промокоды Aroma Butik
Промокоды Яндекс
Путешествия
Постила
Футбол сегодня



```
pip install -e .
```

Чтобы удалить пакет:

```
pip uninstall TinkoffPy
```

И не забудьте указать свой секретный токен доступа в файле `TinkoffPy/Config.py` - без этого запросы к API не будут работать.

Как скачать историю через `Bars.py`

В библиотеке `TinkoffPy` есть скрипт `Bars.py`, который позволяет скачивать исторические данные по выбранным тикерам. Однако перед этим необходимо подготовить список тикеров - его можно скачать через API Московской биржи.

Для этого можно использовать мой Python-скрипт, который возвращает список всех акций, торгующихся в основной секции TQBR Московской биржи:

[SilverFir-TradingBot_backtesting/Backtrader_universal/data-connector/stock ticker search.py at main · empenoso/SilverFir-TradingBot_backtesting](#)

```
# Этот скрипт обращается к Московской бирже (MOEX) по интернету
# и получает список всех доступных тикеров акций на основном
рынке (TQBR).
# Затем он формирует строку вида:
# security_codes = ('SBER', 'GAZP', ...)
# которую можно использовать например в Python-программе скачивания
котировок Bars.py библиотеки TinkoffPy.
```

```
# 24.05.2025
# Михаил Шардин, https://shardin.name/?utm\_source=python
```

```
import requests
```

```
# Делаем запрос к серверу
url = "https://iss.moex.com/iss/engines/stock/markets/shares/boards..."
params = {
    "iss.meta": "off",
    "iss.only": "marketdata",
    "marketdata.columns": "SECID"
}
response = requests.get(url, params=params)
response.raise_for_status() # проверка на ошибку
```

```
# Извлекаем тикеры
data = response.json()
tickers = [item[0] for item in data['marketdata']['data']]
```

```
# Формируем строку
```

```
security_codes_str = f"security_codes = {tuple(tickers)}"  
print(security_codes_str)
```

Результатом будет строка вида `security_codes = ('SBER', 'GAZP', 'LKOH', ...)`, которую можно вставить в [Bars.py](#), чтобы начать загрузку котировок.

Я немного изменил `Examples/Bars.py` и добавил обработку ошибок в `save_candles_to_file` для предотвращения падения скрипта.

Теперь после того как мы подготовили список тикеров с Московской биржи и воспользовались скриптом [Bars.py](#) из библиотеки `TinkoffPy` (или её аналогом для `Alor/Finam`) для скачивания истории, мы получаем набор текстовых файлов.

Каждый полученный файл содержит дневные котировки (OHLCV) для одного тикера. Стандартное именование файлов, например, для данных от Тинькофф, выглядит так: `TQBR.ТИКЕР_D1.txt`. `TQBR` обозначает основной режим торгов акциями на Мосбирже, `ТИКЕР` - код акции (например, `SBER`, `GAZP`), а `_D1` указывает на дневной таймфрейм.

Имя	Тип	Размер	Дата
..		<Папка>	24.05.2025 11:51
TQBR.TRNFP_D1	txt	434 371	24.05.2025 15:16
TQBR.SIBN_D1	txt	416 152	24.05.2025 15:15
TQBR.TATN_D1	txt	401 230	24.05.2025 15:16
TQBR.MSGD_D1	txt	396 764	24.05.2025 15:14
TQBR.MTSS_D1	txt	395 194	24.05.2025 15:14
TQBR.RTKMP_D1	txt	392 711	24.05.2025 15:15
TQBR.TGKB_D1	txt	390 536	24.05.2025 15:16
TQBR.TGKN_D1	txt	390 050	24.05.2025 15:16
TQBR.TGKA_D1	txt	371 945	24.05.2025 15:16
TQBR.ROSN_D1	txt	363 720	24.05.2025 15:15
TQBR.FEES_D1	txt	359 459	24.05.2025 15:12
TQBR.SNGSP_D1	txt	359 296	24.05.2025 15:16
TQBR.TGKBP_D1	txt	359 134	24.05.2025 15:16
TQBR.SBER_D1	txt	355 556	24.05.2025 15:11
TQBR.NVTK_D1	txt	354 234	24.05.2025 15:14
TQBR.RTKM_D1	txt	350 228	24.05.2025 15:15
TQBR.LKOH_D1	txt	345 696	24.05.2025 15:13
TQBR.SNGS_D1	txt	344 991	24.05.2025 15:15
TQBR.CHMF_D1	txt	341 623	24.05.2025 15:12
TQBR.SBERP_D1	txt	340 886	24.05.2025 15:15
TQBR.AFLT_D1	txt	333 909	24.05.2025 15:12
TQBR.PLZL_D1	txt	328 749	24.05.2025 15:15
TQBR.LSRG_D1	txt	327 987	24.05.2025 15:13

Скаченная история

Поиск самых ликвидных акций

Откуда берутся данные: структура файлов

Содержимое файла `TQBR.ТИКЕР_D1.txt` представляет собой таблицу, где колонки разделены табуляцией:

TQBR.ТИКЕР_D1.txt

Здесь:

- **datetime:** дата и время свечи (для дневных данных время обычно 00:00).
- **open:** цена открытия.
- **high:** максимальная цена за день.
- **low:** минимальная цена за день.
- **close:** цена закрытия.
- **volume:** объем торгов в лотах. Важно помнить, что это именно лоты, а не количество акций или денежный объем. Для расчета реального денежного оборота нам потребуется знать размер лота для каждой акции.

Скрипт: находим самые ликвидные акции по кварталам за последние 20 лет

Теперь, когда исторические данные по всем доступным акциям доступны, мы можем приступить к анализу их ликвидности. Для этой задачи я написал Python-скрипт `liquid_stocks_by_quarter.py`. Его основная цель - обработать все скачанные файлы с дневными котировками, рассчитать суммарный торговый оборот в рублях для каждой акции поквартально за указанный период (по умолчанию 5 лет, но в примере ниже будет 20 лет для наглядности в статье) и отобрать топ-N наиболее ликвидных бумаг для каждого квартала. Для статьи отбирал только 7 штук - иначе бы в форматирование не поместилось.

Ключевые шаги работы скрипта:

1. **Поиск файлов:** скрипт сканирует указанную папку (Data/Tinkoff/) на наличие файлов с данными (TQBR.ТИКЕР_D1.txt).
2. **Чтение данных:** каждый файл загружается в pandas DataFrame. Производится парсинг дат и необходимых колонок (OHLCV).
3. **Получение размера лота:** для корректного расчета денежного оборота скрипт пытается подключиться через TinkoffPy к API брокера и получить актуальный размер лота для каждого тикера. Если API недоступно или информация по тикеру отсутствует, используется размер лота равный 1 (это может исказить результаты).
4. **Расчет оборота в рублях:** для каждой дневной свечи вычисляется оборот.

5. **Агрегация по кварталам:** дневные обороты суммируются по каждому тикеру внутри каждого квартала.
6. **Отбор топ-N акций:** для каждого квартала акции сортируются по убыванию суммарного рублевого оборота, и отбирается заданное количество лидеров.
7. **Сохранение результата:** итоговая таблица с самыми ликвидными акциями по кварталам сохраняется в CSV-файл (по умолчанию liquid_stocks_by_quarter.csv).

Скрипт также включает логирование для отслеживания процесса обработки и возможных ошибок, а также отладочный вывод для проверки корректности чтения форматов файлов. Параметры, такие как глубина анализа в годах (years_back) и количество отбираемых акций (top_n_stocks), легко настраиваются.

Вот сам скрипт на Гитхабе, из-за объема (примерно 400 строк) решил в текст статьи не вставлять:

[SilverFir-TradingBot_backtesting/Backtrader_universal/data-connector/liquid_stocks_by_quarter.py at main · empenoso/SilverFir-TradingBot_backtesting](#)

Результат - файл liquid_stocks_by_quarter.csv

Анализируя эти получившиеся данные по самым топ-7 ликвидным акциям с 2005 по 2025 год, можно выделить несколько ключевых периодов:

- 2005-2017: высокая волатильность состава.
Доминировали ТГК-компании (TGKB, TGKA, TGKN) и региональные энергетики.
Состав менялся практически каждый квартал.
- 2018: кардинальная смена лидеров - переход к "голубым фишкам": SBER, GAZP, LKOH, ROSN, GMKN стали основой топа.
- 2018-2025: относительная стабилизация.
Костяк из 4-5 акций (Сбербанк, Газпром, Лукойл) остается неизменным, меняются только 2-3 позиции.

date,quarter,stocks,stocks_count

2005-06-30,2005-Q2,"MSNG,LKOH,SIBN,SNGS,IRKT,RTKM,RTKMP",7

2005-09-30,2005-Q3,"MSNG,LKOH,SIBN,ELFV,IRKT,RTKM,SNGS",7

2005-12-31,2005-Q4,"MSNG,ELFV,LKOH,IRKT,SIBN,SNGS,SBER",7

2006-03-31,2006-Q1,"MSNG,ELFV,LKOH,IRKT,GAZP,RTKM,SIBN",7

2006-06-30,2006-Q2,"MSNG,GAZP,ELFV,LKOH,IRKT,SBER,SIBN",7

2006-09-30,2006-Q3,"MSNG,GAZP,ELFV,LKOH,MSRS,IRKT,ROSN",7

2006-12-31,2006-Q4,"ELFV,MSNG,GAZP,LKOH,UPRO,RTKMP,MSRS",7

2007-03-31,2007-Q1,"TGKB,MSNG,ELFV,UPRO,MSRS,GAZP,ROSN",7

2007-06-30,2007-Q2,"TGKB,TGKN,TGKA,MSNG,ELFV,TTLK,UPRO",7

2007-09-30,2007-Q3,"TGKN,TGKB,TGKA,ELFV,MSNG,TGKBP,OGKB",7

2007-12-31,2007-Q4,"TGKN,TGKB,TGKA,UPRO,ELFV,MSNG,TGKBP",7

2008-03-31,2008-Q1,"TGKB,TGKN,TGKA,TGKBP,ELFV,UPRO,ROSN",7

2008-06-30,2008-Q2,"TGKN,TGKB,TGKA,TGKBP,ELFV,ROSN,OGKB",7

2008-09-30,2008-Q3,"TGKB,TGKN,TGKA,MRKP,TGKBP,UPRO,MSNG",7

2008-12-31,2008-Q4,"TGKB,TGKN,TGKA,MRKP,UPRO,MRKU,MSRS",7

2009-03-31,2009-Q1,"TGKN,TGKB,TGKA,MRKP,TGKBP,ROSN,MRKV",7

2009-06-30,2009-Q2,"TGKN,TGKB,TGKA,MRKP,TGKBP,MSNG,OGKB",7

2009-09-30,2009-Q3,"TGKB,TGKN,TGKA,TGKBP,MSNG,MRKP,OGKB",7

2009-12-31,2009-Q4,"TGKB,TGKN,TGKA,TGKBP,MRKP,TTLK,OGKB",7

2010-03-31,2010-Q1,"TGKB,TGKN,TGKA,MRKY,MRKP,TGKBP,OGKB",7

2010-06-30,2010-Q2,"TGKB,TGKBP,TGKN,TGKA,MRKY,OGKB,UPRO",7

2010-09-30,2010-Q3,"TGKB,TGKA,TGKN,TGKBP,OGKB,MSNG,MRKY",7

2010-12-31,2010-Q4,"TGKB,TGKN,TGKA,TGKBP,MRKV,MRKY,MRKP",7

2011-03-31,2011-Q1,"TGKB,TGKA,TGKN,TGKBP,MRKP,MRKV,MRKY",7

2011-06-30,2011-Q2,"TGKB,TGKA,TGKN,TGKBP,OGKB,UPRO,MRKU",7

2011-09-30,2011-Q3,"TGKB,TGKA,TGKN,TGKBP,OGKB,UPRO,MRKP",7

2011-12-31,2011-Q4,"TGKB,TGKA,TGKN,TGKBP,OGKB,MRKP,MRKU",7

2012-03-31,2012-Q1,"TGKA,TGKB,TGKN,OGKB,TGKBP,ELFV,MSNG",7

2012-06-30,2012-Q2,"TGKB,TGKA,TGKN,OGKB,UPRO,MSNG,TGKBP",7

2012-09-30,2012-Q3,"TGKB,TGKA,TGKN,TGKBP,MRKP,MSNG,OGKB",7

2012-12-31,2012-Q4,"TGKA,TGKB,TGKN,TGKBP,MRKU,MRKP,UPRO",7

2013-03-31,2013-Q1,"TGKN,TGKA,TGKB,UPRO,OGKB,MRKP,TGKBP",7

2013-06-30,2013-Q2,"TGKA,TGKN,TGKB,UPRO,OGKB,TGKBP,MRKP",7

2013-09-30,2013-Q3,"TGKA,TGKB,TGKN,OGKB,UPRO,MRKP,MSNG",7

2013-12-31,2013-Q4,"TGKA,TGKB,MSRS,TGKN,UPRO,TGKBP,OGKB",7

2014-03-31,2014-Q1,"TGKA,TGKN,TGKB,UPRO,MSRS,MRKP,TGKBP",7

2014-06-30,2014-Q2,"TGKA,TGKB,UPRO,TGKN,MRKP,OGKB,TGKBP",7

2014-09-30,2014-Q3,"TGKA,TGKB,TGKN,UPRO,MRKP,TGKBP,OGKB",7

2014-12-31,2014-Q4,"TGKB,TGKA,TGKN,TGKBP,MSRS,UPRO,MRKV",7

2015-03-31,2015-Q1,"TGKA,TGKB,TGKN,TGKBP,UPRO,OGKB,MSNG",7

2015-06-30,2015-Q2,"TGKA,TGKB,OGKB,UPRO,MSNG,TGKN,TGKBP",7

2015-09-30,2015-Q3,"TGKB,TGKA,TGKN,MSNG,TGKBP,UPRO,MRKV",7

2015-12-31,2015-Q4,"TGKB,TGKA,TGKN,UPRO,MSNG,TGKBP,MRKV",7

2016-03-31,2016-Q1,"TGKB,TGKA,UPRO,TGKN,MSNG,TGKBP,MRKP",7

2016-06-30,2016-Q2,"TGKB,TGKA,TGKN,TGKBP,MSNG,MRKP,UPRO",7

2016-09-30,2016-Q3,"TGKN,TGKA,TGKB,TGKBP,UPRO,MRKP,MRKV",7

2016-12-31,2016-Q4,"TGKA,TGKB,TGKN,TGKBP,MSNG,MRKP,OGKB",7

2017-03-31,2017-Q1,"TGKA,TGKB,TGKN,TGKBP,UPRO,PRFN,MRKP",7

2017-06-30,2017-Q2,"TGKN,TGKA,TGKB,MRKP,TGKBP,UPRO,MSNG",7

2017-09-30,2017-Q3,"TGKB,TGKA,TGKBP,TGKN,MRKP,MRKV,MSNG",7

2017-12-31,2017-Q4,"TGKB,TGKA,TGKN,TGKBP,MRKP,MRKV,MRKU",7

2018-03-31,2018-Q1,"TGKA,TGKB,TGKBP,TGKN,MRKP,MRKV,UPRO",7

2018-06-30,2018-Q2,"SBER,GAZP,LKOH,ROSN,GMKN,VTBR,MGNT",7

2018-09-30,2018-Q3,"SBER,GAZP,LKOH,GMKN,MGNT,ROSN,SNGSP",7

2018-12-31,2018-Q4,"SBER,LKOH,GAZP,ROSN,GMKN,MGNT,SNGSP",7

2019-03-31,2019-Q1,"SBER,LKOH,GAZP,GMKN,ROSN,MGNT,ALRS",7

2019-06-30,2019-Q2,"SBER,GAZP,LKOH,GMKN,ROSN,SBERP,MGNT",7

2019-09-30,2019-Q3,"SBER,GAZP,LKOH,GMKN,SNGSP,SNGS,ROSN",7

2019-12-31,2019-Q4,"GAZP,SBER,LKOH,GMKN,SNGS,YDEX,ROSN",7

2020-03-31,2020-Q1,"SBER,GAZP,LKOH,GMKN,ROSN,SNGS,MGNT",7

```
2020-06-30,2020-Q2,"SBER,GAZP,LKOH,GMKN,ROSN,TATN,PLZL",7
2020-09-30,2020-Q3,"SBER,YDEX,GAZP,GMKN,LKOH,PLZL,MGNT",7
2020-12-31,2020-Q4,"SBER,GAZP,LKOH,GMKN,YDEX,PLZL,ROSN",7
2021-03-31,2021-Q1,"SBER,GMKN,GAZP,LKOH,ROSN,YDEX,TATN",7
2021-06-30,2021-Q2,"SBER,GAZP,GMKN,LKOH,VTBR,YDEX,ROSN",7
2021-09-30,2021-Q3,"GAZP,SBER,LKOH,GMKN,ROSN,ALRS,NVTK",7
2021-12-31,2021-Q4,"GAZP,SBER,LKOH,GMKN,ROSN,YDEX,SNGS",7
2022-03-31,2022-Q1,"SBER,GAZP,LKOH,YDEX,GMKN,ROSN,T",7
2022-06-30,2022-Q2,"GAZP,SBER,LKOH,ROSN,GMKN,NVTK,VTBR",7
2022-09-30,2022-Q3,"GAZP,SBER,LKOH,GMKN,NVTK,ROSN,YDEX",7
2022-12-31,2022-Q4,"SBER,GAZP,LKOH,PLZL,GMKN,ROSN,YDEX",7
2023-03-31,2023-Q1,"SBER,GAZP,PLZL,LKOH,ABIO,NVTK,SBERP",7
2023-06-30,2023-Q2,"SBER,GAZP,LKOH,VTBR,MGNT,YDEX,PLZL",7
2023-09-30,2023-Q3,"SBER,LKOH,GAZP,VTBR,TRNFP,MGNT,YDEX",7
2023-12-31,2023-Q4,"SBER,MTLR,LKOH,GAZP,YDEX,SNGSP,MGNT",7
2024-03-31,2024-Q1,"SBER,YDEX,LKOH,T,MTLR,GAZP,VTBR",7
2024-06-30,2024-Q2,"SBER,GAZP,LKOH,T,RNFT,AFKS,YDEX",7
2024-09-30,2024-Q3,"SBER,GAZP,LKOH,YDEX,T,MTLR,NVTK",7
2024-12-31,2024-Q4,"T,SBER,GAZP,LKOH,OZON,YDEX,SMLT",7
2025-03-31,2025-Q1,"SBER,GAZP,T,VTBR,SMLT,LKOH,NVTK",7
2025-05-24,2025-Q2,"GAZP,SBER,T,VTBR,LKOH,NVTK,YDEX",7
```

Этот файл можно использовать для динамического формирования списка торгуемых инструментов в Backtrader на каждый квартал, что позволит адаптировать стратегию к изменяющимся условиям ликвидности на рынке.

7 позиций это конечно мало, но я сделал так специально для статьи. Код открыт - Вы можете провести свои исследования.

Заключение

Итак, мы прошли путь от настройки окружения и загрузки исторических данных с помощью библиотек Игоря Чечета до анализа ликвидности акций Московской биржи. Результат статьи - готовый CSV-файл с топ-7 ликвидными бумагами по кварталам за 20 лет. Это не просто таблица, а инструмент для последующего динамического отбора активов в Backtrader, позволяющий строить более реалистичные и адаптивные торговые стратегии.

Что думаете?

Автор: Михаил Шардин

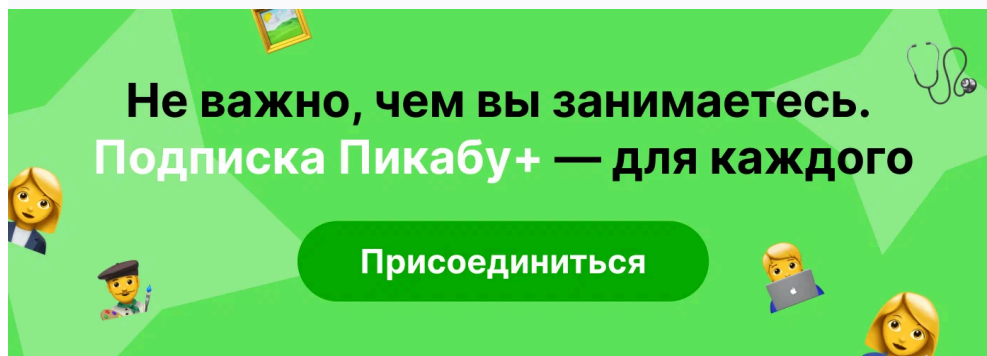
 [Моя онлайн-визитка](#)

 [Telegram «Умный Дом Инвестора»](#)

27 мая 2025 г.



Больше постов читайте по тегу [«Программирование»](#). А если хотите изучить новую профессию, посмотрите актуальные курсы от проверенных школ с реальными отзывами на сайте [Пикабу Курсы](#).



Не важно, чем вы занимаетесь.
Подписка Пикабу+ — для каждого

Присоединиться



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 1 комментарий

Чтобы оставить комментарий, необходимо [зарегистрироваться](#) или [войти](#)

● — ■ —

—
—
—
—
—
—

● — ■ —

—
—
—
—
—
—

● — ■ —

—
—
—
—
—
—

