

**УНИФИКАЦИЯ АВТОМАТИЧЕСКИХ ТРЕЙДИНГОВЫХ СИСТЕМ
С ИСПОЛЬЗОВАНИЕМ ПРИНЦИПА СЕГРЕГАЦИИ ИНТЕРФЕЙСОВ****Бейлак Намаз оглы АЛИЕВ**

аспирант кафедры экономической информатики,
Московский государственный университет имени М.В. Ломоносова,
Москва, Российская Федерация
beylak@yandex.ru
<https://orcid.org/0000-0002-5529-7310>
SPIN-код: 3742-8839

История статьи:

Рег. № 406/2024
Получена 24.06.2024
Получена в
доработанном виде
10.07.2024
Одобрена 20.07.2024
Доступна онлайн
29.08.2024

Специальность: 5.2.2**УДК** 330.47**JEL:** C87, C88, G11**Аннотация**

Предмет. В условиях динамического развития количественных подходов в трейдинге очень важно формировать условия для оперативного тестирования и применения разработанных математических моделей. Модель унификации архитектуры торговых стратегий на основе сегрегированных программных интерфейсов позволит повысить качество апробации исследовательских работ, направленных на развитие количественных торговых стратегий.

Цели. Формирование предложений по унификации архитектуры торговых стратегий для автоматических трейдинговых систем на основе принципа сегрегации интерфейсов.

Методология. Использованы метод эмпирического наблюдения, анализ аналитической и экспертной информации. Применены практики программной инженерии по проектированию систем, в частности, принципы сформированных в виде акронима SOLID.

Результаты. Проведено исследование актуального состояния подходов к разработке архитектуры автоматических трейдинговых систем. Сформированы, разработаны и представлены для обсуждения протоколы торговых стратегий для достижения унификации при разработке торговых стратегий, применяемых в автоматических трейдинговых системах. Результаты работы опубликованы в открытом репозитории и распространяются по лицензии MIT.

Выводы. Развитие и применение единого и открытого репозитория сегрегированных интерфейсов взаимодействия с торговыми стратегиями позволит качественно и оперативно апробировать новые количественные модели. Результаты могут быть интегрированы в деятельность хедж-фондов, использующих количественные стратегии для управления капиталом.

Ключевые слова:

SOLID, ISP, трейдинг,
протокол, торговая
стратегия

© Издательский дом ФИНАНСЫ и КРЕДИТ, 2024

Для цитирования: Алиев Б.Н. Унификация автоматических трейдинговых систем с использованием принципа сегрегации интерфейсов // Финансовая аналитика: проблемы и решения. – 2024. – Т. 17, № 3. – С. 359 – 366.
<https://doi.org/10.24891/fa.17.3.359>

С развитием вычислительных возможностей все более доступными становятся количественные подходы к торговле на рынках ценных бумаг. Согласно различным источникам (см., например, работу [1]), больше половины биржевых операций приходится на автоматические торговые системы (АТС), которыми пользуются не только крупные хедж-фонды,

такие как Renaissance Technologies LLC, но и частные инвесторы. Согласно проведенному анализу развития и роли искусственного интеллекта на фондовой бирже [2], эволюция, цифровизация и глобализация рынка труда в целом и фондовых бирж в частности привели к естественной потребности все больше вовлекать искусственный интеллект в трейдинговую деятельность, не ограничиваясь при этом механическими манипуляциями по открытию или закрытию позиций, по полноценному, нелинейному принятию решений. При этом разрабатываемые автоматические торговые (трейдинговые) системы зачастую представляют собой комплекс сильно связанных подсистем стратегий и механизмов управления заявками. Исследовательские работы, описывающие все больше новых моделей прогнозирования ценных бумаг и мотивация индустрии на применение АТС, выдвигают актуальную потребность в унификации разрабатываемой архитектуры трейдинговых систем.

В работе [3] рассмотрена классическая структура механической торговой системы, а также предлагается авторское видение современной системы. Автор оперирует термином «блоки», подразумевая самостоятельные модули, не уточняя степень изолированности модулей (каждый блок – это выделенный сервер, приложение или поток исполнения кода в рамках одной единой программы). В одной из публикаций¹ наблюдается разделение системы на логические блоки, такие как «Обработчик рыночных данных» (сбор, обработка и хранение рыночных данных), «Менеджер ордеров» (управления и хранения данных о транзакциях), «Стратегия» (модель алгоритмического трейдинга). Также в публикациях встречаются следующие логические блоки: «Управление исполнением», «Риск-менеджмент», «Инфраструктура», «Отчетность и аналитика», «Мониторинг», «Интеграция с внешними системами». Согласно практическому ресурсу², выделяются следующие нефункциональные требования к автоматическим торговым системам:

- 1) масштабируемость;
- 2) производительность;
- 3) модифицируемость;
- 4) отказоустойчивость;
- 5) безопасность;
- 6) надежность;
- 7) аудируемость;
- 8) интероперабельность (функциональная совместимость).

Приведенный список целесообразно демаркировать на инфраструктурные задачи и программно-архитектурные, где первые 5 пунктов зачастую лежат на инфраструктурном уровне, а с 6 пункта – зона ответственности архитектуры программного продукта.

Анализ исследований зарубежных и отечественных авторов приводит к следующему выводу: в экспертном и научном сообществе уделяется внимание архитектуре автоматической торговой системы в целом, взаимодействию крупных подсистем, интеграции с внешними системами. При этом не уделяется внимания архитектуре подсистем, нет дизайна программных модулей, взаимодействия классов, разделения на программные интерфейсы. Отсутствие декомпозиции до уровня программного кода приводит к отсутствию унификации, другими словами, каждый исследователь или практик разрабатывает модели для торговых систем и продумывает индивидуальную архитектуру, а это в свою очередь

¹ Endeavoring to Build a Trading System.

URL: <https://ikequant.substack.com/p/endeavoring-to-build-a-trading-system>

² Martin R.C. Solid Relevance. URL: <https://blog.cleancoder.com/uncle-bob/2020/10/18/Solid-Relevance.html>

затрудняет применение данных разработок из-за необходимости не только изучать саму модель, но и то, как с ней взаимодействовать, усложняется ее подключение к уже имеющейся инфраструктуре.

Наше исследование является попыткой разработки общественного контракта (протокола) унификации автоматических торговых систем путем выработки ряда программных интерфейсов, описывающих полный цикл работы системы. Протоколы (интерфейсы классов), описывающие поведение объектов, с одной стороны, потребуют от исследователей разработки методов взаимодействия с моделями, с другой стороны, разработчики систем смогут вызывать эти модели, не изучая их внутреннюю структуру. Такой подход понизит когнитивную нагрузку для быстрого перехода между моделями или при тестировании новых моделей, что в свою очередь потенциально повысит возможности на валидацию исследовательских моделей.

Основная цель данной работы – формирование предложений по унификации архитектуры торговых стратегий для автоматических трейдинговых систем. Гипотеза исследования: использование принципа сегрегации интерфейсов в архитектуре автоматических трейдинговых систем позволяет улучшить модульность и расширяемость систем. Для достижения поставленной цели необходимо выполнить ряд задач: описать архитектуру автоматизированных трейдинговых систем; сформировать протоколы стратегий и их компонентов; представить сформированные протоколы в виде программной библиотеки на языке общего назначения. Достижение поставленных целей возможно путем соблюдения инженерных практик, сформулированных Робертом Мартином³ в виде акронима SOLID и закладывает пять принципов.

- S (SIP) – принцип единственной ответственности. Принцип предполагает, что класс/модуль/функциональный блок выполняет только единственную задачу. К примеру, если класс отвечает за функцию сохранения документа Excel на диске, то добавление в этот класс функции по формированию этого документа будет нарушением данного принципа и повлечет ограничения по повторному использованию данного функционала.
- O (OCP) – принцип открытости-закрытости. Идеи, заложенные в данный принцип, ведут к достижению высокой надежности кода. Разработанный ранее код протестирован и работает в продуктивной системе, модификация потребует колоссальных ресурсов по повторному тестированию и при этом не даст гарантии отсутствия новых ошибок, в связи с этим требуется закрыть код для модификаций. Следовательно, для добавления нового функционала код должен быть разработан так, чтобы новые функции можно было бы подключать, то есть расширять.
- L (LSP) – принцип Барбары Лисков. Данный принцип со строгой математической формулировкой, представленной Барборой Лисков [4], был перефразирован и включен в SOLID Робертом Мартином. В объектно-ориентированном программировании одной из основных и мощных концепций является механизм наследования. Наследования предоставляют широкие возможности по проектированию программного кода. Данный принцип описывает подход к взаимодействию базовых классов к его наследникам (или основного типа к его подтипам). LSP требует сохранения работоспособности подклассов по принципу базового класса, другими словами, если какая-либо функция выполняет работу с использованием базового класса, то эта же функция без переписывания кода должна без ошибок выполнить работу с использованием подклассов (подтипов).
- I (ISP) – принцип разделения интерфейсов. Данный принцип предполагает разделение интерфейса на множество специфических вместо использования одного большого. Такой подход избавит пользователя интерфейса (функцию) от «лишних» знаний о методах,

³ Martin R.C. Solid Relevance. URL: <https://blog.cleancoder.com/uncle-bob/2020/10/18/Solid-Relevance.html>

которые ему не нужны. Другими словами, для функции $f(x)$ x будет представлен в четко определенном специфическом для интерфейса виде. Вне зависимости от полученного в будущем экземпляра x' функция f будет владеть «знаниями» исключительно о нужных методах.

- D (DIP) – принцип инверсии зависимости. Принцип, направленный на повышение расширяемости программного продукта и его повторного использования. Предполагается, что если функция или класс будут зависеть от абстракций (интерфейсов), то есть не будут ожидать конкретный класс, а только некое свойство (контракт), то в такую функцию можно будет передать любой объект, обладающий свойствами, описанными в интерфейсе.

В ряде публикаций⁴ [5] авторы выдвигают и подтверждают тезис о том, что благодаря использованию SOLID программный продукт достигает возможности повторного использования, расширяемости, простоты и становится сопровождаемым.

Далее разработаем ряд сегрегированных протоколов с учетом принципа ISP (interface segregation principle). В контексте данной работы понятие «протокол» является аналогией программного интерфейса, который описывает требуемое поведение класса реализации, а также представляет механизм структурной типизации в языке Python с версии 3.8, представленной в PEP 544⁵. Использование приведенного принципа и его реализация на языке программирования Python, с одной стороны, подчеркивает ценность для практиков рынка, с другой – открывает возможности для исследователей, разрабатывающих и проверяющих модели, применяемые на рынке ценных бумаг. Рассмотрим следующие компоненты: протокол работы стратегии, протокол прогнозирования цены, протокол определения точек открытия и закрытия позиций, протокол риск-менеджмента, протокол лимитов.

Протокол торговой стратегии. Стратегия должна содержать и предоставлять объекты, являющиеся компонентами стратегии. При работе виртуальной машины автоматической трейдинговой системы (АТС) выполнится загрузка класса стратегии, а также вызовы его методов, указанные в протоколе, и если класс соответствует контракту, то виртуальная машина выполнит задачу без ошибок. Методы стратегии являются исключительно провайдерами объектов, описывающими остальные протоколы. Код протокола представлен в нашем публичном репозитории на GitHub⁶.

Протокол модели прогнозирования движения цены. Исследователи и практики, разрабатывая предиктивные модели, не рассматривают вопрос подключения к торговому терминалу или иному инструменту для выполнения продуктивной деятельности. Зачастую этот вопрос остается на стороне пользователя, который применяет модель. Протокол, представленный в данном случае, является попыткой решить эту проблему. На данный момент описан только один метод, `predict`, принимающий на вход текущий ценовой бар (OHLC) и FIGI (Financial Instrument Global Identifier). Другими словами, исполнитель для получения прогноза произведет вызов данного метода, а в качестве ответа ожидает модель данных, описанную типизированным словарем на листинг кода 1, опубликованный в репозитории.

Листинг кода 1. Модель данных:

```
import typing as tp
from datetime import datetime
```

⁴ Григорьев А.О., Фирсова С.А. Исследование применения принципов объектно-ориентированного программирования SOLID в контексте разработки программно-информационной системы для спортсменов-биатлонистов // Актуальные проблемы науки и образования в условиях современных вызовов: сборник материалов XXIII Международной научно-практической конференции. М.: Печатный цех, 2023. С. 33–46.

⁵ PEP 544 – Protocols: Structural subtyping (static duck typing). URL: <https://peps.python.org/pep-0544/>

⁶ Алиев Б.Н. Securities trading strategies protocols. URL: <https://github.com/beilak/stsp.gi>

```
from decimal import Decimal
class PredictResponse(tp.TypedDict):
    figi: str
    for_date: datetime
    next_price: Decimal.
```

В модели данных ожидается по конкретному инструменту прогноз следующей цены с указанием даты прогноза.

Протокол модели определения точек входа и выхода из позиции. При работе трейдера прогноз цены не всегда является окончательным пунктом принятия решения, для этого предлагается выделить дополнительный класс, задача которого определить удовлетворение условиям открытия/закрытия позиций. В протоколе открытия позиции описан метод `definition_entry_point`, который ожидает FIGI, OHLC, а также модель ответа прогноза. Протокол закрытия позиции аналогичен открытию, но с методом `definition_exit_point`. В качестве результатов работы методов ожидается модель, содержащая цену и объем операции.

Программный класс модели риск-менеджмента стратегии. Вне зависимости от качества проделанной аналитической работы или прогнозной модели управление капиталом невозможно без учета риск-менеджмента, даже незаконная инсайдерская торговля не может дать 100%-ной гарантии. В ситуациях, когда прогнозная модель ошиблась или рынок сменил направление, функционал управления риском позволит сократить убытки.

Виртуальная машина ATC ожидает от объекта реализации трех методов: `is_account_risk_realized`, `is_session_risk_realized`, `is_securities_risk_realized`. Каждый из методов определяет реализацию риска на различных уровнях, уровень всего счета целиком, торговой сессии или отдельно взятого инструмента. В качестве ответа ожидается boolean ответ True/False. В случае True ответа предполагается, что реализовались риски и необходимо запустить механизм обработки риска.

Нами описан ряд протоколов, позволяющих унифицировать моделирование и применение прогнозных моделей в трейдинговой деятельности. Предлагаемый набор протоколов опубликован в публичных репозиториях GitHub и PyPI⁷.

Полный список доступных протоколов на момент написания данной работы: модель работы стратегии (провайдер реализаций компонентов), прогнозирования цены, определение точек открытия и закрытия позиций, риск-менеджмент, лимиты, аккаунт.

В результате нами была принята модель распространения Open Source по лицензии MIT, что, с одной стороны, позволит привнести в работу технологическую независимость, а с другой стороны, предоставить эволюцию продукта через сообщество независимых разработчиков, заинтересованных в унификации публичных методов торговых стратегий для трейдинговых систем.

Подход, предложенный в работе, позволит разработать единую автоматизированную трейдинговую систему, при этом под единой не понимается «в глобальном масштабе». При этом система не предполагает наличия модели торговой стратегии, в такой системе стратегии являются подключаемыми, тем самым обеспечивается возможность оперативно протестировать новые, включить или отключить от работы, не разрабатывая отдельную систему, не разбираясь в деталях модели (воспринимая модель торговой стратегии как черный ящик).

⁷ Алиев Б.Н. Securities trading strategies protocols. URL: <https://github.com/beilak/stsp.gi>;
Алиев Б.Н. STSP. URL: <https://pypi.org/project/stsp/>

Состав протоколов, их методы и модели данных выносятся на обсуждение и свободную модификацию со стороны сообщества. Описанное достоинство полученных результатов работы и успех всего проекта зависит от активности и вовлеченности сообщества, данное условие является критическим. С одной стороны, это позволит достичь независимости и масштаба, с другой стороны, отсутствие вовлеченности сообщества способно остановить развитие проекта на любой из стадий развития.

Нами проанализирован актуальный подход к архитектуре автоматических трейдинговых систем, выделены проблемы отсутствия унификации к дизайну программных продуктов в данной области, выработаны протоколы (контракты) для взаимодействия торговой системы со стратегией и ее компонентами, такими как прогнозная модель, модели открытия/закрытия позиций, риск-менеджмент, лимиты по операциям. Предлагается подход, удовлетворяющий одному из принципов SOLID, в частности, принципу ISP (сегрегация интерфейсов). Выработанные протоколы были разработаны в виде программного пакета на языке Python. Результаты работы опубликованы и распространяются по модели Open Source (лицензия MIT) для формирования сообщества вовлеченных в развитие программных продуктов унифицированных автоматических трейдинговых систем. Открытый репозиторий доступен для модификации. Такой подход позволит вовлечь в обсуждение больше специалистов и оперативно развивать протоколы согласно потребностям и сократить дистанцию между научным и практическим сообществом.

Список литературы

1. Имаев Д.Д., Имаев Д.Х. Моделирование и оптимизация систем автоматической торговли финансовыми инструментами // *Инновации*. 2017. № 2. С. 123–128.
URL: <https://cyberleninka.ru/article/n/modelirovanie-i-optimizatsiya-sistem-avtomaticheskoy-torgovli-finansovymi-instrumentami>
2. Алиев Б.Н. Эволюция рынка труда и роль искусственного интеллекта на фондовой бирже // *Инновации и инвестиции*. 2023. № 10. С. 247–252.
URL: <https://cyberleninka.ru/article/n/evolyutsiya-rynka-truda-i-rol-iskusstvennogo-intellekta-na-fondovoy-birzhe?ysclid=lyr20mqohj656345017>
3. Шумков Е.А. Структуры механических торговых систем // *Прикладная информатика*. 2012. № 3. С. 5–14. URL: <https://cyberleninka.ru/article/n/strukтуры-mehanicheskikh-torgovyh-sistem?ysclid=lyr2jlinsu715364231>
4. Liskov B.H., Wing J.M. A behavioral notion of subtyping. *ACM Transactions on Programming Languages and Systems*, 1994, vol. 16, iss. 6, pp. 1811–1841.
URL: <https://doi.org/10.1145/197320.197383>
5. Madasu V.K., Venna T.V.S.N., Eltaeib T. SOLID Principles in Software Architecture and Introduction to RESM Concept in OOP. *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, 2015, vol. 2, iss. 2.
URL: <https://www.jmest.org/wp-content/uploads/JMESTN42350259.pdf>

Информация о конфликте интересов

Я, автор данной статьи, со всей ответственностью заявляю о частичном и полном отсутствии фактического или потенциального конфликта интересов с какой бы то ни было третьей стороной, который может возникнуть вследствие публикации данной статьи. Настоящее заявление относится к проведению научной работы, сбору и обработке данных, написанию и подготовке статьи, принятию решения о публикации рукописи.

UNIFICATION OF AUTOMATED TRADING SYSTEMS USING THE PRINCIPLE OF INTERFACE SEGREGATION

Beilak N. ALIEV

Lomonosov Moscow State University,
Moscow, Russian Federation
beylak@yandex.ru
<https://orcid.org/0000-0002-5529-7310>

Article history:

Article No. 406/2024
Received 24 Jun 2024
Received in revised
form 10 Jul 2024
Accepted 20 Jul 2024
Available online
29 Aug 2024

JEL Classification:
C87, C88, G11

Keywords: SOLID,
ISP, trading, protocol,
trading strategy

Abstract

Subject. The article discusses a model for unification of trading strategies architecture based on segregated software interfaces, which will improve the quality of testing of research works aimed at the development of quantitative trading strategies.

Objectives. The purpose of the study is to formulate proposals for unification of the architecture of trading strategies for automatic trading systems based on the interface segregation principle.

Methods. The study employs the method of empirical observation, analysis of analytical and expert information, practices of software engineering in systems design, in particular, SOLID design principles.

Results. The study formed, developed and presented for discussion the protocols of trading strategies to achieve unification in the development of trading strategies used in automatic trading systems. The results of the work are published in the open repository and distributed under the MIT license.

Conclusions. The development and application of a single and open repository of segregated interfaces for interaction with trading strategies will enable to qualitatively and quickly test new quantitative models. The findings can be integrated into the activities of hedge funds using quantitative strategies for capital management.

© Publishing house FINANCE and CREDIT, 2024

Please cite this article as: Aliev B.N. Unification of automated trading systems using the principle of interface segregation. *Financial Analytics: Science and Experience*, 2024, vol. 17, iss. 3, pp. 359–366. <https://doi.org/10.24891/fa.17.3.359>

References

1. Imaev D.D., Imaev D.Kh. [Modeling and Optimization of Automated Trading Systems]. *Innovatsii = Innovations*, 2017, no. 2, pp. 123–128. URL: <https://cyberleninka.ru/article/n/modelirovanie-i-optimizatsiya-sistem-avtomaticheskoy-torgovli-finansovymi-instrumentami> (In Russ.)
2. Aliev B.N. [Evolution of the labor market and the role of artificial intelligence in the stock market]. *Innovatsii i investitsii = Innovation and Investment*, 2023, no. 10, pp. 247–252. URL: <https://cyberleninka.ru/article/n/evolyutsiya-rynka-truda-i-rol-iskusstvennogo-intellekta-na-fondovoy-birzhe?ysclid=lyr20mqohj656345017> (In Russ.)
3. Shumkov E.A. [Automated trading systems structures]. *Prikladnaya informatika = Journal of Applied Informatics*, 2012, no. 3, pp. 5–14. URL: <https://cyberleninka.ru/article/n/struktury-mehanicheskikh-torgovykh-sistem?ysclid=lyr2jlnsu715364231> (In Russ.)

4. Liskov B.H., Wing J.M. A behavioral notion of subtyping. *ACM Transactions on Programming Languages and Systems*, 1994, vol. 16, iss. 6, pp. 1811–1841.
URL: <https://doi.org/10.1145/197320.197383>
5. Madasu V.K., Venna T.V.S.N., Eltaeib T. SOLID Principles in Software Architecture and Introduction to RESM Concept in OOP. *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, 2015, vol. 2, iss. 2.
URL: <https://www.jmest.org/wp-content/uploads/JMESTN42350259.pdf>

Conflict-of-interest notification

I, the author of this article, bindingly and explicitly declare of the partial and total lack of actual or potential conflict of interest with any other third party whatsoever, which may arise as a result of the publication of this article. This statement relates to the study, data collection and interpretation, writing and preparation of the article, and the decision to submit the manuscript for publication.