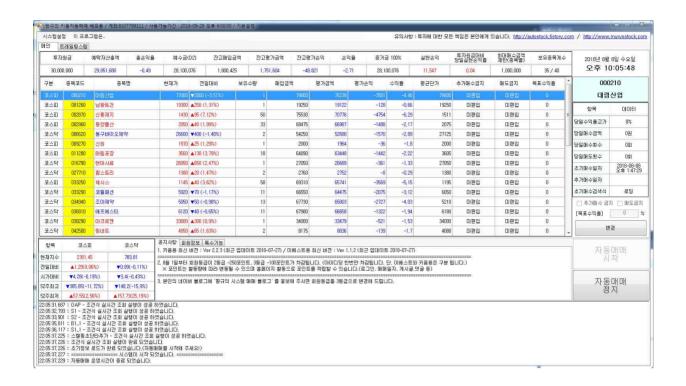


3. Design

주식 자동매매 프로그램





[Revision history]

Revision date	Version #	Description	Author
MM/DD/YYYY	0.00	Type brief description here	Author name
05/05/2023	1.0.0	First Writing	



= Contents =

1.	Introduction
2.	Class diagram
3.	Sequence diagram
4.	State machine diagram
5.	Implementation requirements
6.	Glossary
7.	References



1. Introduction

- Summarize the contents of this document.
- Describe the important points of your design.
- 12pt, 160%.

- Executive Summary

자동매매 프로그램은 사용자의 지정한 조건에 따라 자동으로 주식, 가상화폐 등을 매매하는 프로그램으로, 수익을 극대화하고 투자자의 시간과 노력을 절약할 수 있습니다. 본 프로그램은 높은 수익을 추구하는 투자자들에게 특히 유용합니다.

- Business Goals

자동매매 프로그램은 사용자의 수익 극대화를 목적으로 합니다. 프로그램은 사용자가 지정한 조건에 따라 자동으로 매매를 수행하며, 높은 수익을 얻을 수 있는 적절한 시점에 매수/매도를 수행합니다. 또한, 사용자의 투자 금액을 최적화하여 투자 리스크를 최소화하도록 설계되었습니다.

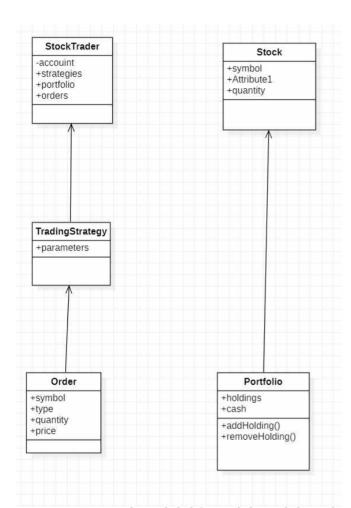
- Technical Goals

자동매매 프로그램은 안정적인 서비스 제공을 목표로 합니다. 이를 위해, 프로그램은 사용자의 투자 계좌에 직접 접근하지 않고, 외부 API를 통해 정보를 가져오며, 매매는 사용자의 승인을 거쳐 수행됩니다. 또한, 매매 알고리즘은 빠르고 정확한 실행을 위해 최적화되어야 하며, 사용자 인터페이스는 직관적이고 이해하기 쉬운 디자인으로 구현되어야 합니다. 프로그램의 보안성 또한 강화되어야 하며, 오류 발생 시 신속한 대처가 가능하도록 모니터링 시스템을 구축해야 합니다.



2. Class diagram

- Draw a class diagram.
- Describe each class in detail (attributes, methods, others) (table type).
- 12pt, 160%.



StockTrader: 주식 트레이딩을 수행하는 메인 클래스입니다. 주식 계정, 트레이딩 전략, 포트폴리오 및 주문과 관련된 정보를 포함합니다.

Stock: 개별 주식을 나타내는 클래스입니다. 주식의 심볼, 가격, 수량 등의 정보를 포함합니다.

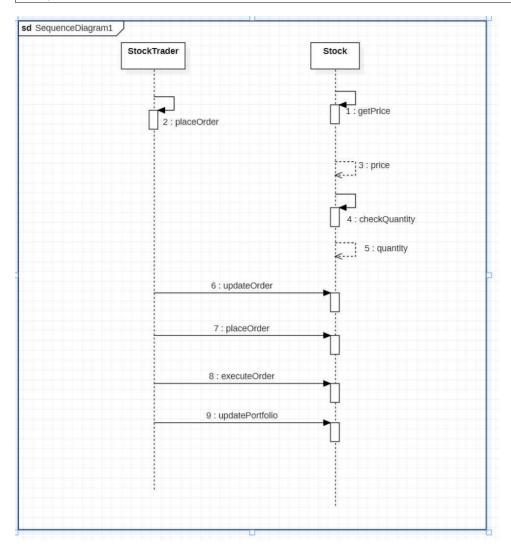
TradingStrategy: 주식 트레이딩 전략을 나타내는 클래스입니다. 전략에 대한 매개 변수 등을 포함합니다.

Order: 주식 주문을 나타내는 클래스입니다. 주문의 심볼, 유형, 수량, 가격 등의 정보를 포함합니다. Portfolio: 트레이딩 포트폴리오를 나타내는 클래스입니다. 보유 주식, 현금 등의 정보와 보유 주식을 추가하거나 제거하는 메서드를 포함합니다.



3. Sequence diagram

- Draw sequence diagrams for the whole functions of your system (this is related to the use cases you made in the Conceptualization phase).
- Explain each sequence diagram.
- 12pt, 160%.



StockTrader 객체는 placeOrder() 메서드를 호출하여 주문을 생성합니다.

Stock 객체는 getPrice() 메서드를 호출하여 주식의 가격을 가져옵니다.

Stock 객체는 checkQuantity() 메서드를 호출하여 주식의 수량을 확인합니다.

StockTrader 객체는 updateOrder() 메서드를 호출하여 주문을 업데이트합니다.

StockTrader 객체는 placeOrder() 메서드를 호출하여 주문을 실행합니다.

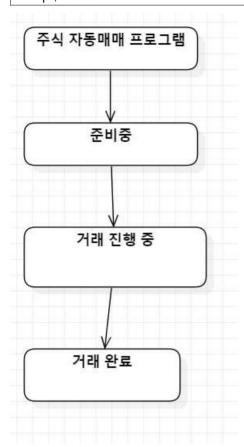
StockTrader 객체는 executeOrder() 메서드를 호출하여 주문을 실행합니다.

StockTrader 객체는 updatePortfolio() 메서드를 호출하여 포트폴리오를 업데이트합니다.



4. State machine diagram

- Draw state machine diagrams for the client and the server system.
- Explain each state machine diagram.
- 12pt, 160%.



주식 자동매매 프로그램의 세 가지 상태를 보여줍니다. 초기 상태는 "준비중"이며, 프로그램이 시작될 때이 상태에서 시작합니다. 이후에는 "거래 진행 중" 상태로 전환되어 실제 주식 거래가 이루어지는 동안 프로그램이 실행됩니다. 마지막으로, 거래가 완료되면 "거래 완료" 상태로 전환됩니다.



5. Implementation requirements

- Describe operating environments to implement your system.
- 12pt, 160%.

프로그래밍 언어: 주식 자동매매 프로그램을 구현할 프로그래밍 언어를 선택합니다. 주식 시장 데이터 처리 및 통합을 위해 Python, Java, C++, C# 등과 같은 언어를 고려할 수 있습니다.

개발 환경: 선택한 프로그래밍 언어에 적합한 개발 환경을 설정합니다. 이는 IDE (통합 개발 환경) 또는 텍스트 편집기, 컴파일러, 빌드 도구 및 버전 관리 시스템 등의 설치를 포함할 수 있습니다.

데이터 소스: 주식 자동매매 프로그램에 필요한 데이터 소스를 식별합니다. 실시간 주식 시장 데이터, 과 거 가격 데이터, 회사 재무 데이터, 뉴스 피드, 경제 지표 등을 포함할 수 있습니다. 이러한 데이터를 액 세스하고 검색하기 위해 사용할 API 또는 데이터 제공 업체를 결정합니다.

거래 플랫폼 통합: 주식 자동매매 프로그램이 특정 거래 플랫폼이나 증권사와 상호작용해야 한다면 해당 플랫폼에서 지원하는 API나 프로토콜을 확인합니다. 선택한 프로그래밍 언어와의 호환성을 확인하고 거래 플랫폼의 기능에 액세스하기 위한 필요한 권한 및 인증 메커니즘을 평가합니다.

알고리즘적 거래 전략: 프로그램이 구현할 알고리즘적 거래 전략을 정의합니다. 기술적 분석 지표, 수학적 모델, 머신 러닝 알고리즘 등을 활용할 수 있습니다. 이러한 전략을 효과적으로 구현하기 위해 필요한 라 이브러리나 프레임워크를 확인합니다.

리스크 관리 및 주문 실행: 프로그램에 구현할 리스크 관리 규칙과 주문 실행 메커니즘을 정의합니다. 손실 제한 주문, 이익 실현 주문, 포지션 크기 조절, 리스크-보상 비율 등과 같은 기능을 고려합니다. 거래 시스템에서의 리스크 관리를 위한 규정 요구 사항과 모범 사례를 준수합니다.

오류 처리 및 로깅: 예외, 네트워크 연결 오류, 데이터 오류 등의 상황에 대비하여 오류 처리 및 로깅 메 커니즘을 구현합니다. 오류 발생 시 적절한 조치를 취하고, 로그를 기록하여 추후 분석 및 디버깅에 활용 할 수 있도록 합니다.

백테스트 및 성능 분석: 구현한 주식 자동매매 프로그램의 성능을 평가하기 위해 백테스트를 수행합니다. 과거 데이터를 사용하여 프로그램을 실행하고 성과를 측정하며, 최적화 및 개선을 위한 분석을 수행합니다. 다.

보안 및 데이터 개인 정보 보호: 주식 자동매매 프로그램이 다루는 데이터의 보안과 개인 정보 보호에 대한 사항을 고려합니다. 데이터 암호화, 접근 제어, 보안 프로토콜 및 개인 정보 보호 규정을 준수해야 합니다.



6. Glossary

- Specifically describe all of the terms used in this documents.
- 12pt, 160%.

로그인 (Login): 시스템에 접속하기 위해 사용자가 입력한 계정 정보를 검증하는 과정입니다.

자동로그인 (Auto Login): 사용자가 최초 로그인 후, 이후에는 로그인 정보를 저장하여 다음 접속 시 자동으로 로그인하는 기능입니다.

매매정보 업데이트 (Update Trading Information): 주식 가격, 거래량, 등의 정보를 실시간으로 수집하여 시스템 내에 업데이트하는 기능입니다.

매매 전략 생성 (Create Trading Strategy): 사용자가 설정한 매매 규칙과 조건에 따라 자동매매 프로그램이 매매를 수행하는 전략을 생성하는 기능입니다.

매매 실행 (Execute Trading): 생성된 매매 전략을 바탕으로 자동매매 프로그램이 주식을 매수하거나 매도하는 기능입니다.

매매내역 조회 (Check Trading History): 자동매매로 이루어진 주식 거래 내역을 조회하는 기능입니다. 자동매매 설정 (Configure Automated Trading): 매매 규칙, 거래금액, 손익분기점 등 자동매매에 필요한 설정을 변경하거나 추가하는 기능입니다.

7. References

- Describe all of your references (book, paper, technical report etc).
- 12pt, 160%.

주식 시장을 이기는 마법의 자동매매 블로그에서 사진을 첨부하였습니다