

클래스 명세서

1. 도메인 계층

아래 클래스들은 주식 시장의 비즈니스 로직을 담당함

클래스명	책임(역할)	주요 필드
Instrument	모든 종목(현물/선물/ETF)의 공통 인터페이스 및 데이터 제공	symbol, name, currentPrice, highPrice, lowPrice, changeRate, InstrumentType
SpotStock	일반 주식(현물) 종목	(Instrument 상속, 추가 필드 없음)
FutureContract	선물 종목 (기초자산 + 레버리지 기반 가격 계산)	shared_ptr<Instrument> underlying, double leverage
CustomETF	사용자 커스텀 ETF (여러 종목 + 비율 조합)	vector<Component> components (Component: 종목 + weight)
Market	전체 종목 리스트 관리, 랜덤 가격 갱신, 상승률 상위 N개 종목 선택	vector<shared_ptr<Instrument>> instruments
Position	계좌가 보유한 특정 종목의 포지션	shared_ptr<Instrument> instrument, int quantity, double avgPrice
Account	계좌(현금 + 포지션 보유) 관리	double cashBalance, vector<Position> positions
Order	실행된 주문(체결) 기록	shared_ptr<Instrument> instrument, int quantity, double price, OrderSide side
OrderManager	매수/매도 주문 처리, 계좌 업데이트, 주문 내역 저장	Account& account, Market& market, vector<Order> history
RiskManager	리스크/청산 로직 담당• 손실 80% 시 선물 전부 강제 청산• 만기 지난 선물 자동 청산	(필드는 사용자의 설계에 따라 추가, 일반적으로 Account·Market 참조됨)

2. UI 계층

사용자 입력과 콘솔 출력 담당

클래스명	책임(역할)	주요 필드
Application	프로그램 전체 라이프사이클 관리, 화면 전환, 메인 루프 관리	Market, Account, OrderManager, unique_ptr<Screen> currentScreen
Screen	모든 화면의 추상 베이스 클래스	Application* app

MainScreen	메인 화면 (시간, 잔고, 상위 5개 종목, 메뉴 출력)	상속받은 app
AllStocksScreen	전체 종목 리스트 정보 출력	상속받은 app
HoldingsScreen	보유 종목(현물/선물/ETF) 출력	상속받은 app
TradeScreen	매수/매도 메뉴 → 종목 선택 → 수량 입력 → 주문 실행	상속받은 app
CustomETFScreen	사용자 커스텀 ETF 구성/수정	상속받은 app

3. 유тиl 계층

부가적인 도우미 클래스

클래스명	책임(역할)	주요 메서드
TimeUtil	현재 시간 문자열 반환	static std::string now()
RandomUtil	난수·가격 변동률 생성	static double randomRate(double min, double max)

메서드 기능 설명

Application

- **Application()**
→ Market, Account, OrderManager, RiskManager를 초기화한다.
- **run()**
→ tick 실행 → 위험관리 수행 → 화면 출력 순으로 메인 루프를 반복한다.
- **changeScreen(Screen* next)**
→ 현재 화면을 다른 화면(Screen 객체)으로 교체한다.
- **exit()**
→ 프로그램 실행을 종료한다.
- **initInstruments()**
→ 초기 Spot/ETF/선물 종목들을 Market에 등록한다.

Screen (추상 클래스)

- **show()**
→ 화면 출력 및 사용자 입력 처리를 담당하는 순수 가상 함수.

MainScreen

- **show()**
→ 현재 시각, 잔고, 상승률 상위 5종목, 메뉴를 출력하고 사용자 입력을 처리한다.

AllStocksScreen

- **show()**
→ Market의 전체 종목 목록을 출력한다.

HoldingsScreen

- **show()**
→ 보유 중인 현물/ETF/선물 포지션 리스트를 출력한다.

TradeScreen

- **show()**
→ 심볼/수량/타입을 입력받아 매수 또는 매도 주문을 실행한다.

CustomETFScreen

- **show()**
→ 구성 종목과 비율을 입력받아 커스텀 ETF를 생성하고 Market에 등록한다.
-

Instrument

- **updatePrice(double newPrice)**
→ 현재가 업데이트 후 고가/저가/변동률을 갱신한다.
- **recalcPrice()**
→ 파생상품 가격 재계산(SpotStock은 별도 변화 없음).

SpotStock

- **SpotStock(...)**
→ 심볼, 이름, 초기 가격을 받아 기본 주식 객체를 생성한다.

FutureContract

- **recalcPrice()**
→ 기초자산 가격 × 레버리지로 선물 가격을 계산한다.
- **isExpired(std::string today)**
→ 오늘 날짜가 만기일을 초과했는지 확인한다.

CustomETF

- **addComponent(inst, weight)**
→ ETF 구성 종목에 (종목+비율) 하나를 추가한다.
- **clearComponents()**
→ ETF 구성 요소를 모두 제거한다.
- **recalcPrice()**
→ 각 구성 종목의 가격 × 비율의 합으로 ETF 가격을 재계산한다.

Market

- **addInstrument(inst)**
→ Spot/Future/ETF 종목을 목록에 등록한다.
- **findBySymbol(symbol)**
→ 해당 심볼을 가진 Instrument를 반환한다.
- **tick()**
→ 모든 Spot 가격을 랜덤 변동시키고 Future/ETF 가격을 자동 재계산한다.
- **getTopMovers(n)**
→ 변동률이 높은 상위 n개 종목을 반환한다.

Account

- **getEquity(market)**
→ 현금 + 보유 포지션 평가금액 합계를 반환한다.
- **buy(inst, qty, price)**
→ 종목 매수 후 잔고 및 포지션을 업데이트한다.
- **sell(inst, qty, price)**
→ 종목 매도 후 잔고 및 포지션을 업데이트한다.
- **getPositionsByType(type)**
→ 특정 타입(현물/선물/ETF)의 포지션만 필터링하여 반환한다.

Position

(메서드 없음 — 단순 데이터 구조)

Order

- **Order(...)**
→ 주문의 시간, 가격, 수량, 방향(매수/매도)을 저장하는 레코드를 생성한다.

OrderManager

- **placeMarketOrder(symbol, type, qty, side)**
→ 시장가 매수/매도 주문을 실행하고 Account에 반영한다.
- **getHistory()**
→ 전체 주문 이력을 반환한다.

RiskManager

- **checkFuturesExpiry(today)**
→ 만기일이 지난 선물 포지션을 자동 청산한다.
- **checkForcedLiquidation()**
→ 계좌 Equity가 초기 자산의 20% 이하일 경우 선물 포지션을 전부 강제 청산한다.

TimeUtil

- **now()**
→ 현재 시각을 "HH:MM:SS" 형태로 반환한다.
- **todayDate()**
→ 오늘 날짜를 "YYYY-MM-DD" 형식으로 반환한다.

RandomUtil

- **randomRate(min, max)**

→ min~max 사이의 랜덤 실수(주가 변동률)를 생성한다.