

프로그래밍 역량 강화 전문기관, 민코딩

Trading System



TDD + Mocking Project

Trading System

private 에 대한 Unit Test는 생략하자.

✓ private 는 Unit Test 하지 않아도 된다.

- private 는 public에 의해 수행된다.
- public의 Unit Test 수행해도, 대부분의 private은 Test가 가능하다.

✓ private를 Test 하고 싶다면?

- 모두 public으로 변경하여 Test를 수행한다.
- Test가 끝난 함수는 private로 변경한다.

제공되는 API

✓ Nemo증권과 Kiwer증권사에서 제공되는 주식거래 API

- 사내 Github : <https://github.samsungds.net/cra1-sec/TradingSystem>
- 사외 Github : <https://github.com/mincoding1/TradingSystem>

✓ KiwerStock

- login(ID, PASS)
- buy(종목코드, 수량, 가격)
- sell(종목코드, 수량, 가격)
- currentPrice(종목코드)

✓ NemoStock

- certification(ID, PASS)
- purchasingStock(종목코드, 가격, 수량)
- sellingStock(종목코드, 가격, 수량)
- getMarketPrice (종목코드, 시간)



개발해야 하는 Application - 1

✓ 자동 매매 프로그램

- 증권사 선택 기능 : selectStockBroker()
 - 키워 or 네모 증권 선택 가능
- 로그인 기능 : login(id, pass)
- 매수 기능 : buy(종목코드, 가격, 수량)
- 매도 기능 : sell(종목코드, 가격, 수량)
- 현재가 확인 기능 : getPrice(종목코드)

개발해야 하는 Application - 2

✓ 자동 매매 프로그램

▪ 기능 1 : buyNiceTiming(종목, 금액)

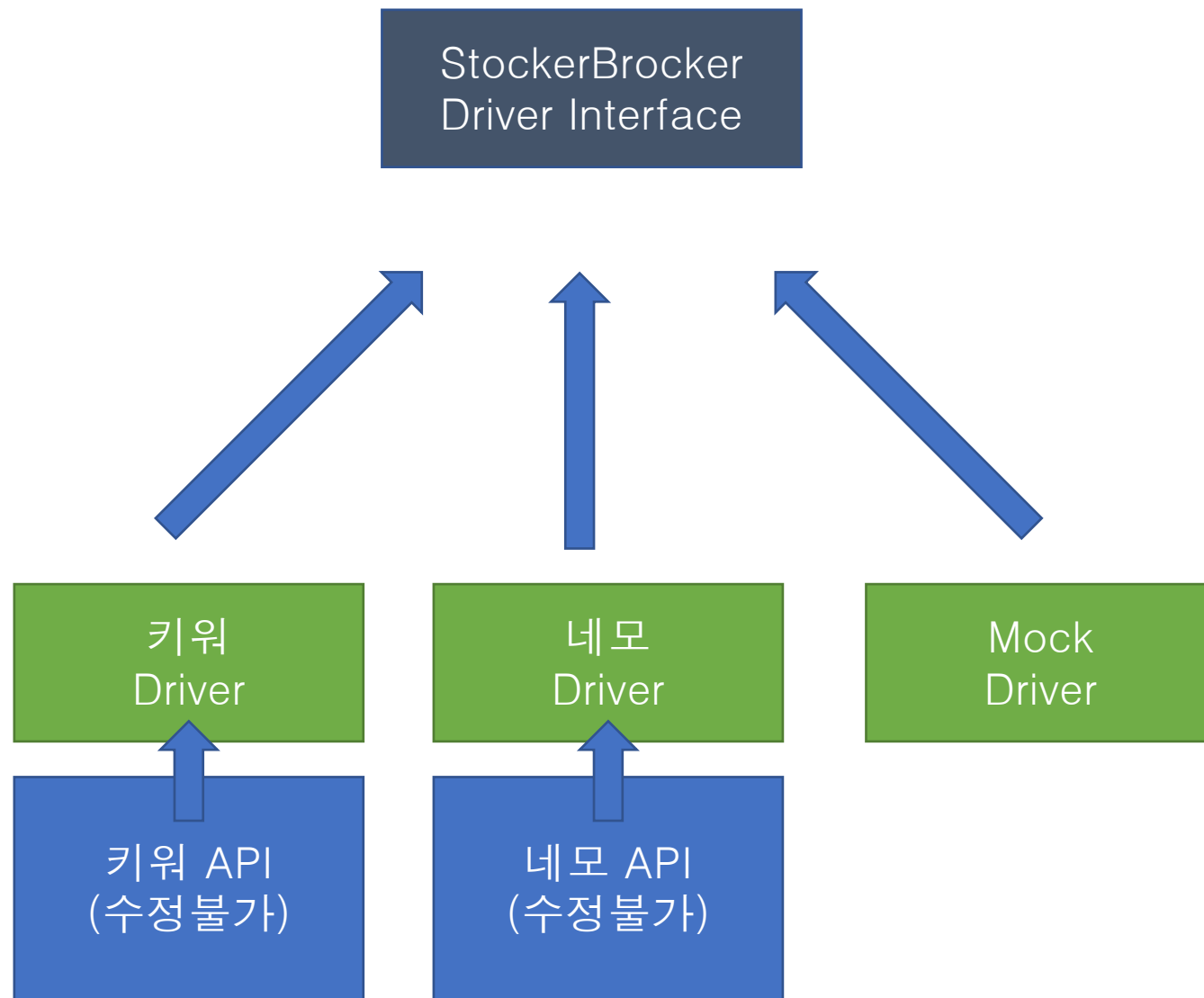
- 가격이 올라가는 추세라면,
사용자가 걸어놓은 금액에 최대 수량만큼 매수한다.
- 현재가 기준으로 매수한다.

▪ 기능 2 : sellNiceTiming(종목, 수량)

- 가격이 내려가는 추세라고 판단되면,
사용자가 설정한 수량만큼 주식을 모두 매도한다.
- 현재가 기준으로 매도한다.

Stock Brocker

- ✓ OCP로 개발 필요
 - 확장성을 고려한 개발



[참고] 핑퐁 페어프로그래밍

✓ 핑퐁 페어프로그래밍

- 출처 : <https://anthonysciamanna.com/2015/04/18/ping-pong-pair-programming.html>

Loop {

- 개발자 A : 실패하는 테스트를 작성
- 개발자 B : 통과하기에 충분한 코드만 작성하여, 테스트 통과 후 리팩토링
- 개발자 B : 다음 실패하는 테스트 작성
- 개발자 A : 통과하기에 충분한 코드만 작성하여, 테스트 통과 후 리팩토링

}

Unit Test를 작성한다는 것은
한 명의 개발자가
다른 개발자에게 명확히 할 일을 부여한다는 의미.

진행방법

✓ 팀장님

- 할일(ToDo)를 작성한다.
- 스켈레톤 코드조차 하지 않는다.

✓ 팀원분들

- 메신저에 올라온, ToDo 중 하나를 맡아 개발한다.

구체적인 진행방법

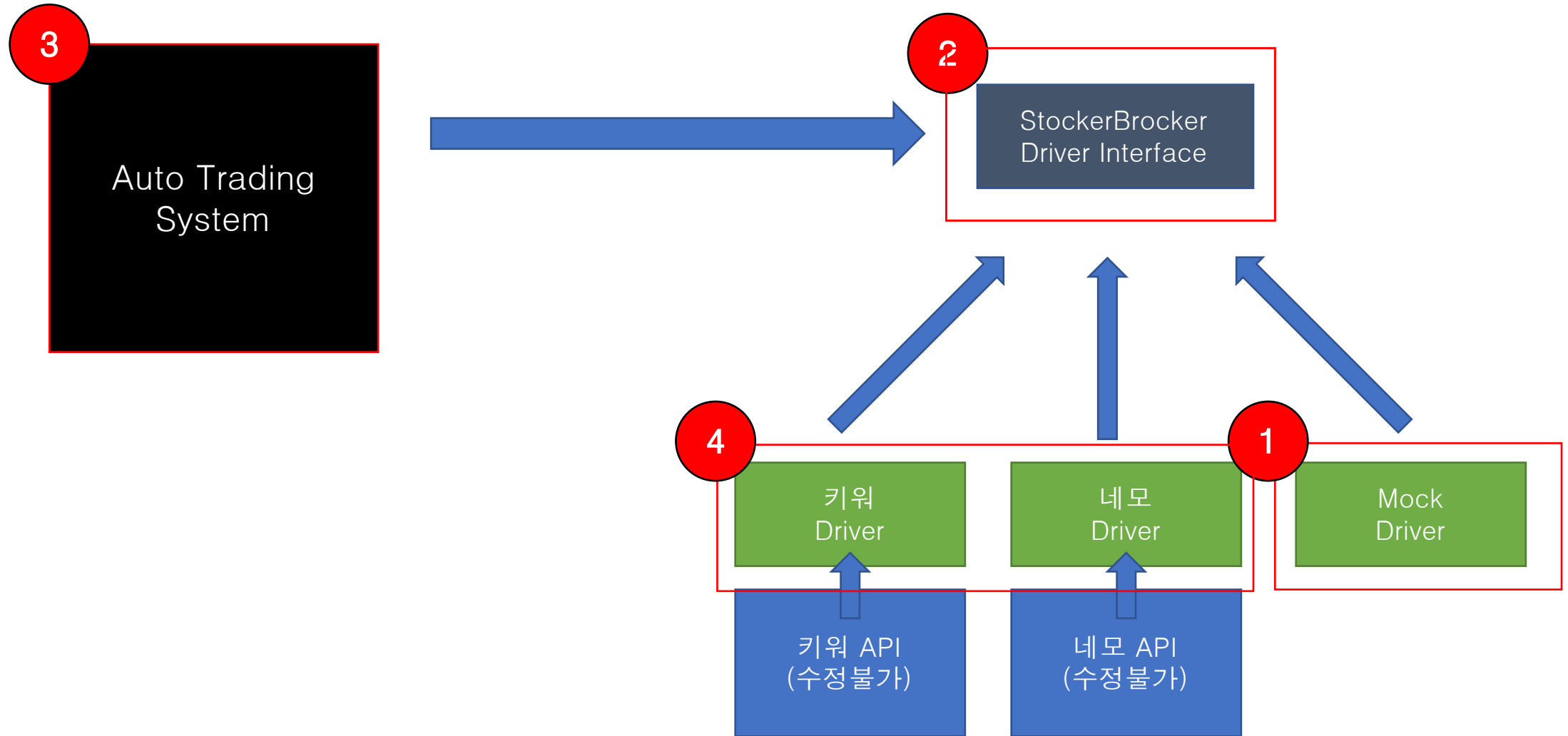
✓ 팀장님

- Unit Test Branch 생성
- 설계 & Unit Test 작성
- 코드리뷰

✓ 팀원분들

- A : Branch 생성, Pass 되도록 개발 + Refactoring, PR 요청
- B : Branch 생성, Pass 되도록 개발 + Refactoring, PR 요청
- C : Branch 생성, Pass 되도록 개발 + Refactoring, PR 요청

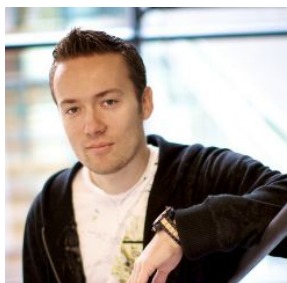
개발 순서



TDD RIP 사건을 통해, TDD 더 살펴보기

끝으로

[참고] TDD의 장단점 토론



- ✓ **TDD is dead** – Ruby on Rails 개발자 (David Heinemeier Hansson)
 - <https://www.moreagile.net/2014/05/IsTDDdead1.html>
 - TDD는 Test에 대해 눈뜨게 해주는 좋은 역할이었다.
 - Unit Test에만 중요성이 너무 높게 치중되어있다.
느리지만 더 많은 모듈을 한꺼번에 테스트하는 System Test도 중요하다.



- ✓ 그 이후, 켄트백, 마틴파울러와의 미팅
 - <https://jinson.tistory.com/271>