iOS Seminar 4

한상현

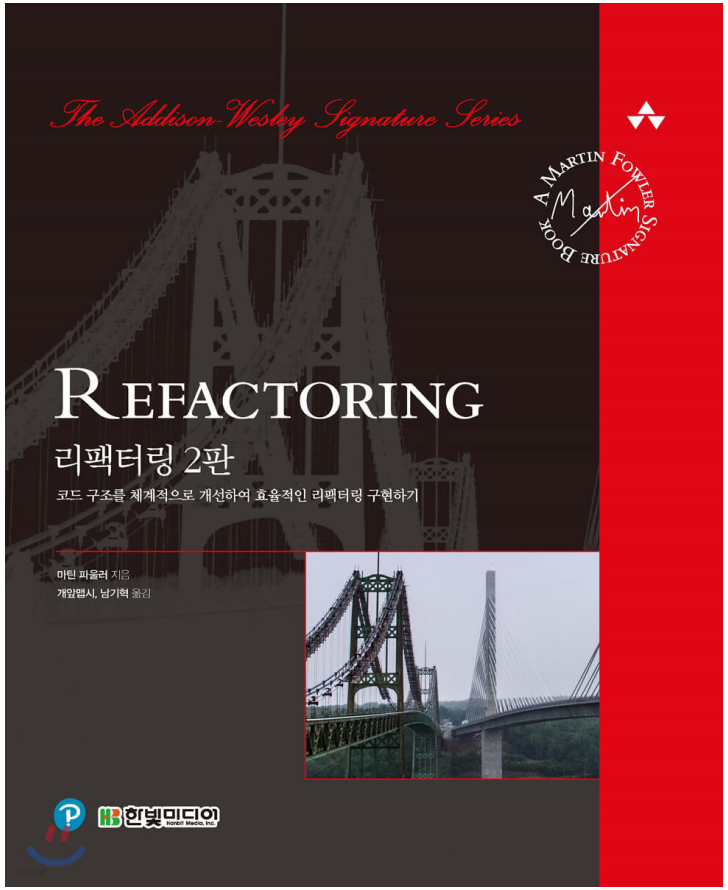
오늘 배울 내용

- 과제3 피드백
- 리팩토링에 대하여
- 구조는 왜 중요한가?
- 의존성이란? 의존성 주입과 그 역전 원칙? 테스트의 필요성? TDD?
- 함께 리팩토링을 해봅시다

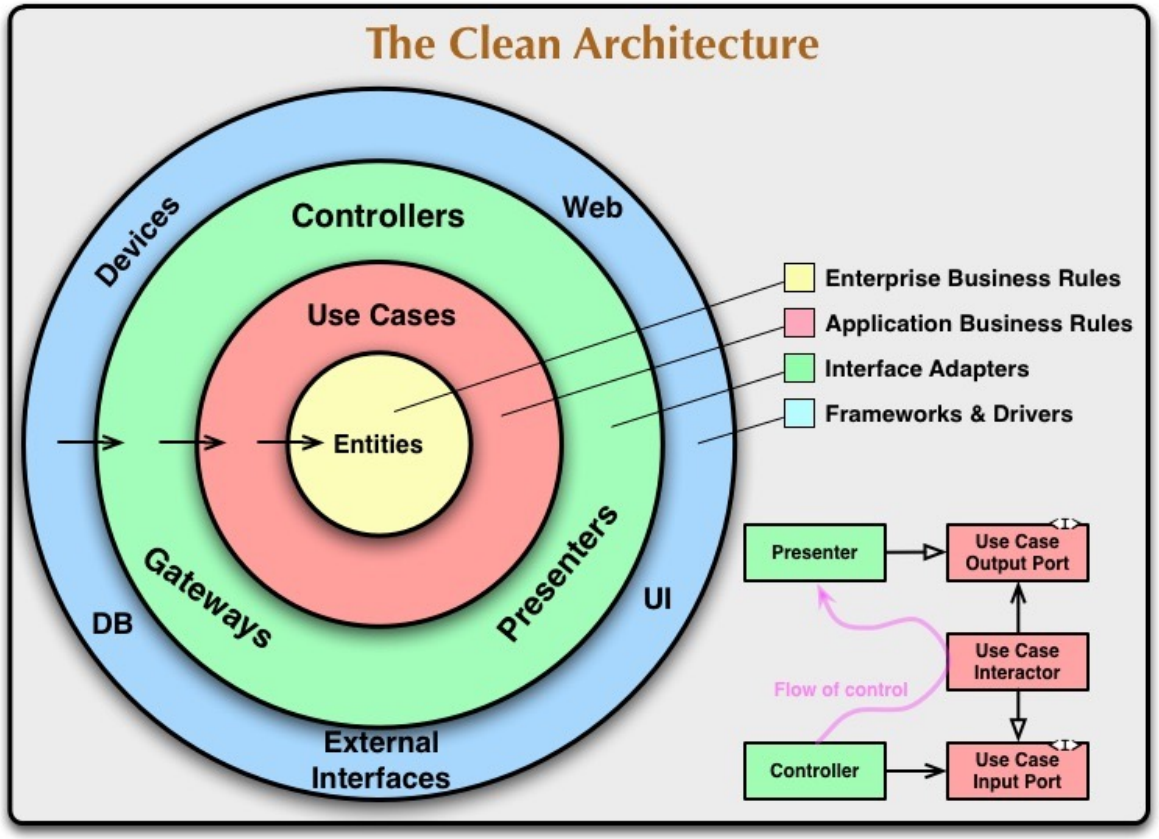
과제3 피드백

- 명령형 프로그래밍에 익숙한 우리들.. 어쩔 수 없습니다
- 구조를 잘 잡지 못하겠는 우리들.. 이것도 어쩔 수 없죠
- 그래도 다들 기능 자체를 잘 구현해주셔서 굉장히 수고하셨다는 말씀드립니다

리팩토링 (Refactoring)



구조


The Clean Architecture
Devices
Controllers
Web
Use Cases
Entities
Gateways
Presenters
DB
UI
External
Interfaces
Enterprise Business Rules
Application Business Rules
Interface Adapters
Frameworks & Drivers
Presenter
Use Case
Output Port
Use Case
Interactor
Flow of control
Controller
Use Case
Input Port

의존성이란?

- “A가 B를 의존한다.” -> 의존대상 B가 변하면, 그것이 A에 영향을 미친다.

Ex)

햄버거 가게 요리사는 햄버거 레시피에 의존한다.

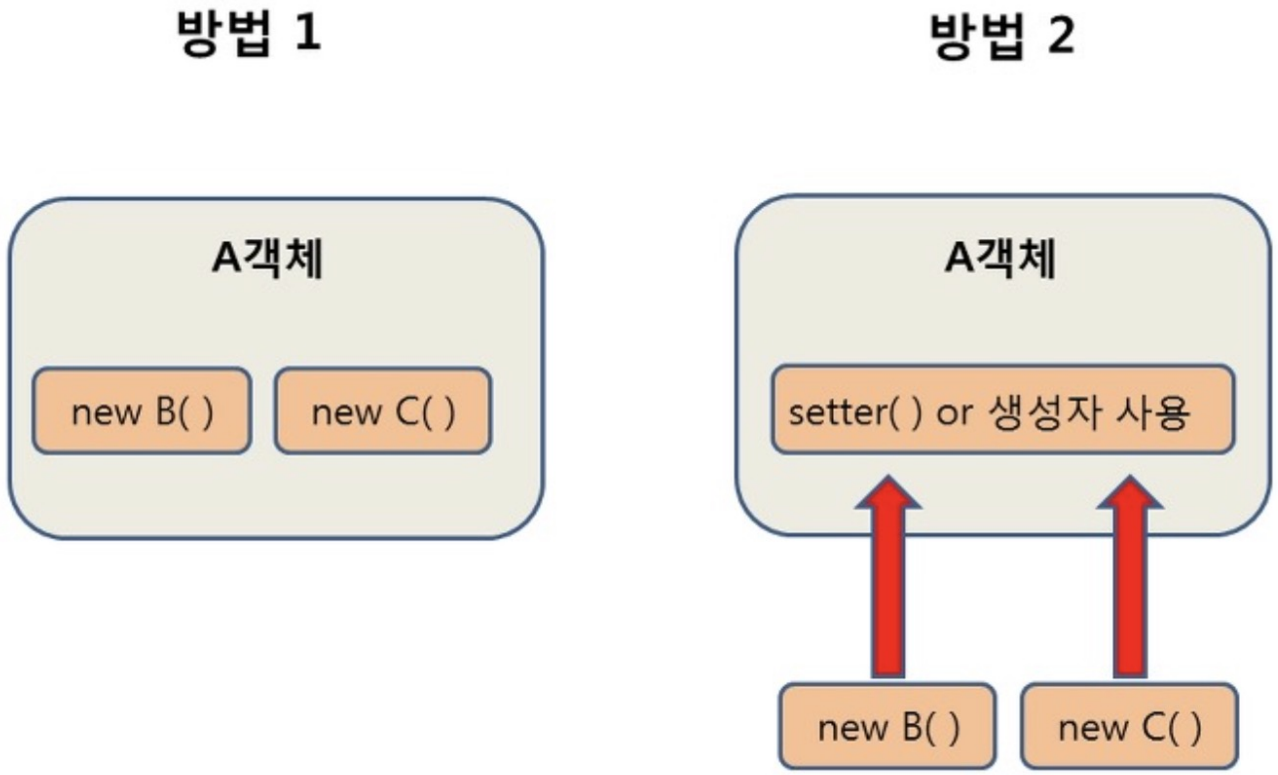
햄버거 레시피가 변화하게 되었을 때, 변화된 레시피에 따라서

요리사는 햄버거 만드는 방법을 수정해야 한다.

레시피의 변화가 요리사의 행위에 영향을 미쳤기 때문에,

“요리사는 레시피에 의존한다”고 말할 수 있다.

의존성 주입 (Dependency Injection : DI)



의존관계 역전 원칙 (Dependency Inversion Principle)

- SOLID (객체지향 설계)의 5가지 원칙 중 하나

- A가 B를 의존할 때, B는 실제 객체가 아닌 이를 추상화한 인터페이스 (protocol) 이어야 한다.

- B protocol을 실제로 구현하는 저수준 객체 (더 세부적이고 사소한(?) 기능을 하는 객체는 자주 바뀌고, 이 변화에 맞춰 A를 바꾸려면 힘이 들기 때문에 A를 바꾸지 않고 B를 적당히 변화시켜 추상화만 만족하는 방식으로 구조 유지가 가능하다.

Rx + MVVM

- ViewController

  1) Component들의 디자인

  2) Component들의 constraint 세팅 -> 오토레이아웃

  3) VM에 유저 액션, View Lifecycle에 맞춰 작업 요청

  4) VM으로부터 데이터를 구독 or 수신하여 유저에게 노출

Rx + MVVM

- ViewModel

  1) VC에 작업 요청을 받는 input 메소드

  2) VC에 결과를 전달하는 output variable

  3) 라우팅 로직, 결과값 수신을 위한 router와 listener, interator, presenter 등을 들고 있을 수 있음

Rx + MVVM

- Domain : Usecase, Manager

  1) VM에서 요청하는 네트워크 데이터 요청, 캐시 데이터 요청, 전역적으로 사용되는 데이터 등을 관리하고 있음

  2) 상황에 맞춰 VM에서 데이터 요청 메소드를 콜하면 적정 수준까지 wrapping하여 전달

- Repository

  - 서버나 캐시로부터 데이터를 직접 요청하여 Return 해주는 객체

함께 리팩토링해보기

- 과제2 코드를 같이 바꿔보면서 감을 잡아봅시다!
- Rx를 사용할게요

Assignment 4 : 영화 소개 앱 리팩토링

- 여러분들이 과제3에서 구현한 영화 앱을 리팩토링합니다.
- 반드시 MVVM + Router + Usecase + Repository의 형태로 리팩토링하셔야 합니다.
- 의존성은 반드시 주입되어야 하고, 의존성 역전은 필수는 아닙니다 (생각보다 작업량이 많고 헷갈려서)

- 구체적인 스펙은 체크리스트의 형태로 제공하겠습니다. (이번엔 엄밀하게 팩트 위주로 작성해서 헷갈리지 않게 하겠습니다.)
- 과제 기한 : 11월 14일 월요일 오후 11시 59분
  (이전 과제 페널티로 이른 제출 필요하신 분 예외 : 각자 과제3 코멘트 참조하셔서 확인 바랍니다)