

웹 서버 구축

5주차 스몰과제

- MVC, MVP, MVVM -

빅데이터전공

20185135

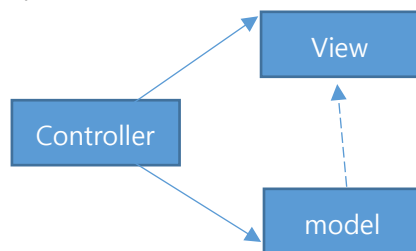
서주현

● MVC 패턴

1. 정의

: Model, View, Controller로 구성된 디자인 패턴

2. 패턴의 구조



3. 패턴의 구조에 기반한 동작 설명

: 입력이 Controller에 들어옴

-> Controller에서 Model에 명령 보내어 Model 상태 변경

-> Controller는 Model을 나타낼 View 선택

-> View는 Model을 이용하여 화면을 나타냄

4. 특징

: Controller는 여러 개의 View를 선택할 수 있는 1:n 구조

Controller는 View를 선택할 뿐 직접 업데이트 할 수 없다.

5. 장점 & 단점

장점 : 가장 단순하여 보편적으로 많이 사용되는 디자인 패턴이다.

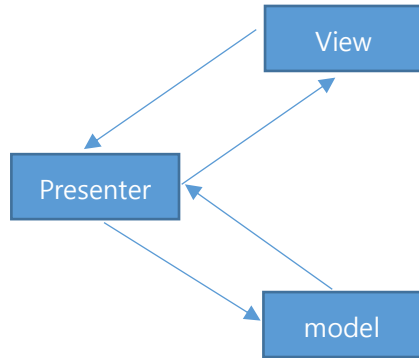
단점 : View와 Model 사이의 의존성이 높아 어플리케이션이 커질수록 복잡해지고 유지보수가 어렵다.

● MVP 패턴

1. 정의

: Model, View, Presenter로 구성된 디자인 패턴

2. 패턴의 구조



3. 패턴의 구조에 기반한 동작 설명

: 입력이 View에 들어옴

-> View가 데이터를 Presenter에 요청

-> Presenter는 Model에 데이터 요청 -> Model에서 데이터 응답

-> Presenter는 View에 데이터 응답

-> View는 Presenter가 응답한 데이터로 화면 나타냄

4. 특징

: Presenter는 View와 Model의 인스턴스를 가지고 있어 둘을 연결하는 역할을 한다.
Presenter와 View는 1:1 관계이다.

5. 장점 & 단점

장점 : View와 Model 사이의 의존성이 없어 MVC 패턴의 단점을 해결

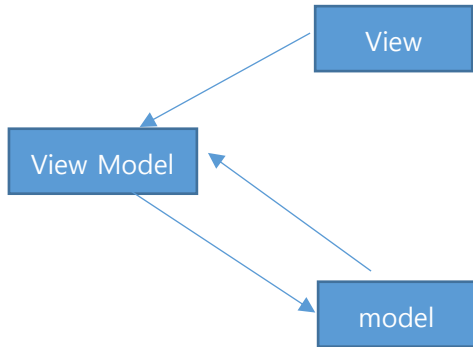
단점 : View와 Presenter 사이의 의존성이 높고 복잡해질수록 의존성이 강해진다.

● MVVM 패턴

1. 정의

: Model, View, View Model로 구성된 디자인 패턴

2. 패턴의 구조



3. 패턴의 구조에 기반한 동작 설명

: 입력이 View를 통해 들어옴

-> Command 패턴으로 View Model에서 입력을 전달

-> View Model은 Model에 데이터 요청 -> Model은 요청받은 데이터를 응답

-> View Model은 응답받은 데이터를 가공 저장

-> View는 View Model과 Data Binding하여 화면 나타냄

4. 특징

: Command패턴과 Data Binding을 사용하여 구현

View Model과 View는 1:n 관계

5. 장점 & 단점

장점 : View와 Model 사이의 의존성이 없고 Command패턴과 Data Binding을 사용하여 View와 View Model의 의존성이 없다. 독립적이어서 모듈화 하여 개발 가능

단점 : View Model의 설계가 어렵다.