

MVC, MVP, MVVM

1. 위 디자인 패턴을 자신이 이해한 대로 서술하고 왜 이렇게 다양한 패턴이 탄생했을까 의견을 제시

- **MVC** : Model View Controller의 약자로 애플리케이션을 세가지의 역할로 구분한 개발 방법론이다. 사용자가 Controller를 조작하면 Controller는 Model을 통해서 데이터를 가져오고 그 정보를 바탕으로 시각적인 표현을 담당하는 View를 제어해서 사용자에게 전달하게 된다.
- **MVP** : MVC 패턴과 다르게 입력이 View에서 처리된다. Presenter는 View의 인스턴스를 갖고 있으며 View와 1대1 관계이고, 그에 해당하는 Model의 인스턴스 또한 갖고 있기때문에 View와 Model 사이에서 다리와 같은 역할을 한다. View에서 이벤트가 발생하면 Presenter에게 전달해주고 Presenter는 해당 이벤트에 따른 Model을 조작하고 그 결과를 바인딩을 통해 View에게 통보를 하여 View를 업데이트 시켜준다.
- **MVVM** : ViewModel 뷰모델 말그대로 View를 나타내주기 위한 Model이라고 생각하면 된다. View보다는 Model과 유사하게 디자인되며, View의 바인딩 될 때 가장 강력하다. MVP와 같이 View에서 입력이 처리된다. Command를 통하여 Behavior를 View의 특정한 ViewAction(Event)와 연결할 수 있으며, ViewModel의 속성과 특정 View의 속성을 Binding 시켜 줌으로써 ViewModel 속성이 변경 될때마다 View를 업데이트 시켜줄 수 있다.

