**스몰과제\_5**

**20175105\_곽영주**

**MVC, MVP, MVVM 아키텍처 패턴 조사**

1. MVC
2. 정의

Model, View, Controller의 합성어이며, 데이터 처리와 비즈니스 로직, 사용자 화면을 각각 독립적으로 개발하는 디자인 패턴이다.

1. 패턴의 구조

Model: 애플리케이션에서 사용되는 데이터를 처리하는 부분이다.

View: 사용자 인터페이스 부분이다.

Controller: 사용자의 입력처리와 흐름 제어하는 부분이다.

1. 패턴의 구조에 기반한 동작 설명

* Controller로 사용자의 입력이 들어온다.
* Controller에서 입력을 확인 후, Model 업데이트한다.
* Controller에서 Model의 데이터를 보여줄 View 선택한다.
* View는 Model을 이용하여 화면에 나타낸다.

1. 특징

* View-Controller 관계는 One-to-Many 관계이다.
* Controller는 Model과 View의 브리지 역할을 한다.

1. 장점 & 단점

장점: 각각의 패턴들을 구분하여 개발하므로 유지보수가 용이하며, 유연성과 확장성이 높다.

단점: View와 Model이 서로 의존성이 높아서 애플리케이션이 커질수록 복잡하고 유지보수가 어렵다.

1. MVP
2. 정의

Model, View, Presenter의 합성어이며, Model과 View를 분리하여 Presenter를 통해서만 데이터를 전달받아 개발하는 디자인 패턴이다

1. 패턴의 구조

Model: 애플리케이션에서 사용되는 데이터를 처리하는 부분이다.

View: 사용자 인터페이스 부분이다.

Presenter: View에서 요청한 정보를 Model로부터 가공해서 View로 전달하는 부분이다.

1. 패턴의 구조에 기반한 동작 설명

* View로 사용자 입력이 들어온다.
* View는 Presenter에게 데이터를 요청한다.
* Presenter는 Model에게 데이터를 요청한다.
* Model은 Presenter에서 요청한 데이터를 응답한다.
* Presenter는 View에서 요청한 데이터를 응답한다.
* View는 Presenter가 응답한 데이터를 이용하여 화면을 나타낸다.

1. 특징

* View-Presenter 관계는 One-to-One 관계이다.
* Presenter는 View와 Model을 연결하는 역할을 한다.
* MVC 패턴의 문제점을 해결할 수 있다.

1. 장점 & 단점

장점: Presenter를 통해서만 데이터를 전달받기 때문에 Model-View의 의존성이 낮다.

단점: MVC 패턴의 View-Model의 의존성은 해결했지만, View-Presenter의 의존성이 높다.

1. MVVM
2. 정의

Model, View, ViewModel의 합성어이며, 서로 간의 의존성을 분리하여 화면 처리에 초점을 맞춰 개발하는 디자인 패턴이다.

1. 패턴의 구조

Model: 애플리케이션에서 사용되는 데이터를 처리하는 부분이다.

View: 사용자 인터페이스 부분이다.

ViewModel: View를 표현하기 위해 만든 Model이며, View를 나타내기 위한 데이터 처리를 하는 부분이다.

1. 패턴의 구조에 기반한 동작 설명

* View로 사용자 입력이 들어온다.
* View에서 입력을 확인 후, Command 패턴으로 ViewModel에 명령한다.
* ViewModel은 Model에게 데이터를 요청한다.
* Model은 ViewModel에서 요청한 데이터를 응답한다.
* ViewModel은 응답 받은 데이터를 가공하여 저장한다.
* View는 ViewModel과 Data Binding하여 화면에 나타낸다.

1. 특징

* View-ViewModel 관계는 One-to-Many 관계이다.
* Command와 Data Binding으로 구현되었다.
* MVC, MVP 패턴에서 나타난 View와 의존성 문제점을 해결할 수 있다.

1. 장점 & 단점

장점: Command와 Data Binding으로 MVP 패턴과 달리 View와의 의존성을 완벽히 분리한다.

단점: 규모가 큰 애플리케이션을 위해서 고안된 디자인 패턴인 만큼, 작은 애플리케이션에서 사용하면 오버헤드가 커진다.