

## #02. Design Pattern(디자인 패턴)

전미정

### 1. 디자인 패턴(Design Pattern)

: 디자인 패턴이란 반복적으로 발생하는 문제 해결을 위한 재사용 가능한 구조화된 방식이다. 주어진 문제를 어떻게 해결할 것인가에 대해 이미 많은 경험을 했던 사람들이 해결방식을 구조화 해놓은 것으로 다양한 종류가 있다.

클래스의 기능과 역할을 분리/그룹화하여 객체지향적인 프로그램 설계를 가능하게 한다. 각각의 클래스(그룹)은 서로의 수정에 영향을 주지 않으며 이러한 설계는 손쉬운 유지보수(수정), 캡슐화, 코드의 재사용을 가능하게 한다. 디자인 패턴을 사용할 때에는 패턴의 사용 목적, 장단점, 약점등에 대해 충분히 이해를 해야지 발생할 수 있는 문제점을 예측 할 수 있다.

디자인 패턴은 이미 증명된 개발방법을 이용함으로써, 개발 속도를 향상시켜 줄 수 있지만 개발 목적에 잘 부합하지 않은 패턴을 선택하면 복잡성만 증가시킬 수 있으므로 주의해서 사용해야한다.

### 2. Singleton pattern

: 어플리케이션 전 영역에 걸쳐 특정 클래스에서 단 한번의 인스턴스를 생성하는 것으로 전역변수와 비슷해 static 이라 할 수 있다. 공통적으로 사용되는 부분, 다수의 객체를 관리하기 할 때, 내용이 변경되면 다른 클래스에도 변경 내용이 적용되어야 할때 사용된다.