프레임워크(Framework)

프레임워크란 : Frame + work

프레임워크 = 확장성이 있는 기반 코드(디자인패턴)+ 확장에 필요한 라이브러리

기반 코드는 성능, 보안, 확장성, 안정성을 만족함

소프트웨어의 구체적인 부분에 해당하는 설계와 구현을 재사용이 가능하게끔

일련의 협업화된 형태로 클래스들을 제공하는 것

=> 시스템의 하부 구조를 구현하는데 들어가는 노력을 절감해줌

사용 이유

비기능적인 요소들을 초기 개발마다 구현해야 하는 불합리성

이를 해결함으로써 기능적인 요구사항에 집중

디

프레임워크의 구성요소

IoC(Inversion of Control)

인스턴스의 생성과 소멸까지의 생명주기 관리를

개발자가 아닌 컨테이너가 대신 해줌

=> 일반적인 프로그램 흐름과 반대로 동작하므로 IoC라고 함

컨테이너 역할을 해주는 프레임워크에 권한을 넘겨서 개발자의

코드가 신경써야 할 것을 줄이는 전략

클래스 라이브러리

특정 부분의 기술적인 구현을 라이브러리 형태로 제공

디자인 패턴

좋은 코드를 만들기 위한 일종의 가이드

프로그램 개발에서 자주 나타나는 이슈를 해결하기 위한 방법

개발과정에서 발견된 Know-How를 축적하여 이름을 붙여 재사용하기 위한

좋은 형태로 패턴, 규약을 묶어서 정리한 것

Design Patterns : Elements of Resuable Object-Oriented Software에서 유래

사용 이유

소스코드 변경을 최소화

범용적인 코딩 스타일 적용하기 위한 적용

직관적인 코드 사용

프레임워크와 라이브러리의 차이

라이브러리 : 자주 사용되는 로직을 재사용하기 편리하도록 정리한 일련의 코드 집합

FW : Library

프레임워크가 개발자코드를 호출 ⇔ 개발자코드에서 Lib를 호출

프레임워크 클래스를 서브클래싱 ⇔ 독립적으로 작성

프레임워크가 제어 ⇔ 유저가 제어

SW 재사용성을 높이는 방안

Copy & Paste

초보적인 방식, 비슷한 예에서 소스에서 복사해 옴

JDK 버전이 바뀌면 사용한 클래스를 모두 변경해야 함

메서드 호출

유사한 기능들을 모아 메서드로 정의하여 재사용, 자주 사용됨

하나의 메서드에 여러 클래스들이 영향을 받음

클래스 상속

메서드 호출과 동일하지만, 메서드가 변경되도 인터페이스가

변하지 않으면 클래스 영향X

부모 클래스가 변하면 영향 받음

AOP(Aspect Oriented Programming)

AOP가 핵심관리모듈의 코드를 직접 건드리지 않고 필요한 기능만 동작하도록

하는데는 위빙(weaving)이라고 하는 특수한 작업이 필요하다.

AOP는 OOP를 더욱 OOP답게 만들어 줄 수 있다.