

8/29 정리

스프링

스프링은 좋은 객체 지향 애플리케이션을 개발할 수 있게 도와주는 프레임워크
객체를 설계할 때 역할(인터페이스)와 구현을 명확히 분리

좋은 객체 지향 설계의 5가지 원칙

SOLID

클린코드로 유명한 로버트 마틴이 좋은 객체 지향 설계의 5가지 원칙을 정리

- SRP: 단일 책임 원칙(single responsibility principle)
- OCP: 개방-폐쇄 원칙 (Open/closed principle)
- LSP: 리스코프 치환 원칙 (Liskov substitution principle)
- ISP: 인터페이스 분리 원칙 (Interface segregation principle)
- DIP: 의존관계 역전 원칙 (Dependency inversion principle)

SRP 단일 책임 원칙

-한 클래스는 하나의 책임만 갖는다

-변경이 있을 때 파급효과가 적으면 단일 책임 원칙을 잘 따른것이다

OCP 개방-폐쇄 원칙

-소프트웨어 요소는 확장에는 열려 있고 변경에는 닫혀 있어야 한다

-클라이언트 코드를 변경하지 않고 구현 객체를 변경할 수 있어야 한다

LSP 리스코프 치환의 원칙

-객체는 프로그램의 정확성을 깨뜨리지 않으면서 하위 타입의 인스턴스로 바꿀 수 있어야한다

Ex) 자동차 인터페이스의 엑셀은 앞으로 가는 기능, 뒤로 가면 lsp 위반, 느리더라도 앞으로 가야한다.

ISP 인터페이스 분리 원칙

-하나의 큰 범용 인터페이스 보다 세분화된 여러 개의 인터페이스가 낫다

Ex) 자동차 인터페이스 -> 운전 인터페이스 정비 인터페이스

DIP 의존관계 역전 원칙

-역할(인터페이스)에 의존하게 해야한다. 그래야 유연하게 구현체를 변경할 수 있다.

다형성 만으로는 OCP, DIP를 지킬 수 없다.

때문에 스프링의 DI 컨테이너가 필요하다.