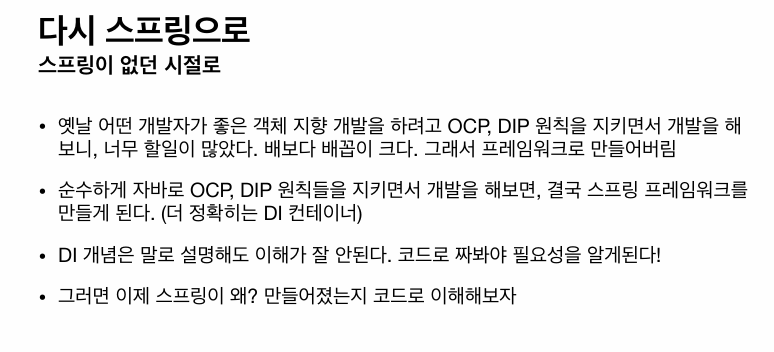
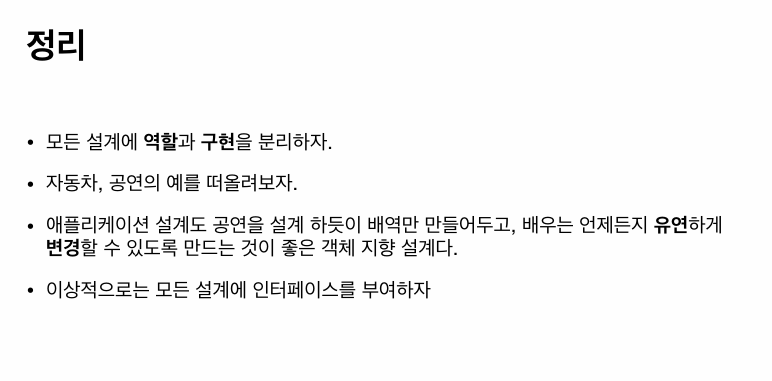


스프링은 다형성과 OCP DIP를 가능하게 지원

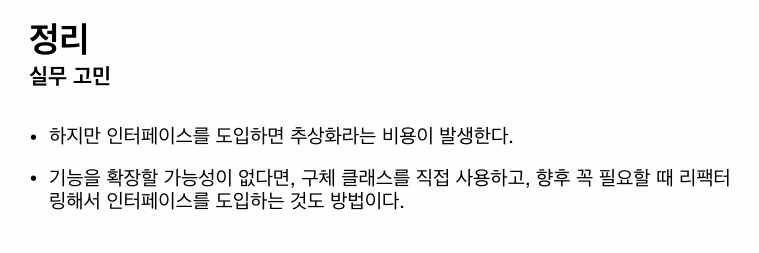
DI 컨테이너 : 자바의 객체들을 어떤 컨테이너 안에 넣어놓고 이 안에서 의존관계를 연결하고 주입해주는 역할을 함. -> 클라이언트의 교체 없이 클라이언트 코드 변경없이 기능을 확장할 수 있게 된다.





인터페이스를 먼저 설계하고 구현체를 나중에 구현하는 방식으로 개발하면, 프로토타입 개발을 진행한 후 실제 규격이 정해지면 차후에 구현체 교체를 통해 코드를 바꿀 수 있다.

하부 구현 기술들에 대한 선택을 최대한 미룰 수 있는 장점이 있음.



추상화가 돼버리면 개발자가 코드를 한 번 더 열어봐야한다. 인터페이스만으로는 실제 구현 클래스가 뭔지 알 수 없다. 추상화가 존재하면 코드변경을 쉽게할 수 있는 장점이 있지만, 그만큼 구현체를 이해하는데 좀 더 많은 시간 소요가 필요하다. 기능 확장할 가능성이 없으면 구체 클래스를 쓰고, 향후에 필요하면 리팩터링해서 인터페이스를 도입하는 것이 중요하다.

역할과 구현을 나눠서 순수하게 인터페이스와 객체를 자바로만 구현해볼 예정.

요구사항이 변경되었을 때 잘 변경가능한가를 확인할 것임.