1. IoC(Inversion of Control)

의존성에 대한 제어권은 원래 자기 자신에게 있었지만 스프링에서는 자기 자신 이외의 누군가가 객체를 주입해준다. (`누군가 내 생성자에 주입해 주겠지` 하는 생각)

1. IoC 컨테이너 – ApplicationContext(BeanFactory)

직접 쓸 일은 많지 않다. 실제로 보통 코드에서 보이지 않는다.

빈(객체)을 만들고 엮어주며 제공해준다.

오로지 빈만 관리할 수 있다. 의존성을 관리해 준다.

예) 리파지토리로 생성된 객체(빈)을 찾아서 컨트롤러의 생성자에 주입해준다.

IOC 컨테이너에서 빈을 꺼내서 사용하는게 @Autowired 이다.

애노테이션으로 컨트롤러, 컴포넌트, 서비스, 리파지토리가 붙어 있으면 빈이다.

1. 빈(Bean)

스프링 IoC 컨테이너가 관리하는 객체

* 등록 방법
  + Component Scanning
    - @Component
      * @Repository
      * @Service
      * @Controller

애노테이션 자체는 주석과 같다. 기능이 없다.

애노테이션을 마커로 사용해서 애노테이션을 처리하는 프로세서들이 있는 것이다.

@Component Scan을 처리하는 프로세서가 있다.

스프링 프로젝트의 기본 클래스가 속한 패키지 아래의 컴포넌트만 찾아서 빈으로 등록해 준다.