1) 자바를 이용한 DI 설정 방식의 2가지 주요 애노테이션을 설명하시오.

Spring3.0 버전부터 애노테이션을 이용하여 빈과 관련된 정보를 설정할 수 있게 되었다. XML 설정 없이 자바 코드를 이용해서 빈 객체 생성과 빈 객체 간의 의존 관계 설정까지 할 수 있다. 하나의 클래스 안에 복수의 빈을 등록할 수도 있다. @Bean 애노테이션과 메서드 이름을 이용해서 컨테이너가 사용할 빈 객체를 생성한다. 자바 설정 방식에서는 빈 객체를 직접 생성한다. @Bean 메서드를 불러들여서 객체를 취득한다. XML에서는 <property> 태그나 <constructor-arg> 태그를 이용해서 설정하였으나 자바 설정에서는 직접 의존 객체를 주입해야 한다.

1) - ① @Configuration

빈 설정 메타 정보를 담고 있는 클래스를 선언한다. 해당 클래스가 스프링 설정으로 사용된다.

1) - ② @Bean

클래스 내의 메서드를 정의하여 새로운 빈 객체를 정의할 때 사용한다. name 속성을 사용하여 새로운 빈 이름을 적용할 수 있다.

2) 빈 객체 라이프 사이클에서 빈을 생성 후 초기화하는 방법과 빈 종료 전 전처리 과정을 수행할 수 있는 방법을 4가지를 정리하시오.

빈 객체의 라이프사이클은 초기화 -> 이용 -> 종료 3단계로 진행되고, 빈 생성 후 초기화 작업과 빈 종료 전 전처리 과정을 수행할 수 있는 방법을 제공한다.

2) - ① XML 기반

초기화 작업 : <bean> 요소의 init-method 속성에 메서드 지정

전처리 작업 : <bean> 요소의 destroy-method 속성에 메서드 지정

2) - ② 애노테이션 기반

초기화 작업 : @PostCOnstruct 애노테이션이 붙은 메소드를 선언

전처리 작업 : @PreDestroy 애노테이션이 붙은 메소드를 선언

2) - ③ 자바 기반

초기화 작업 : @Bean의 initMethod 속성에 메소드를 지정하고 그 메소드를 호출

전처리 작업 : @Bean의 destroyMethod 속성에 메소드를 지정하고 그 메소드를 호출

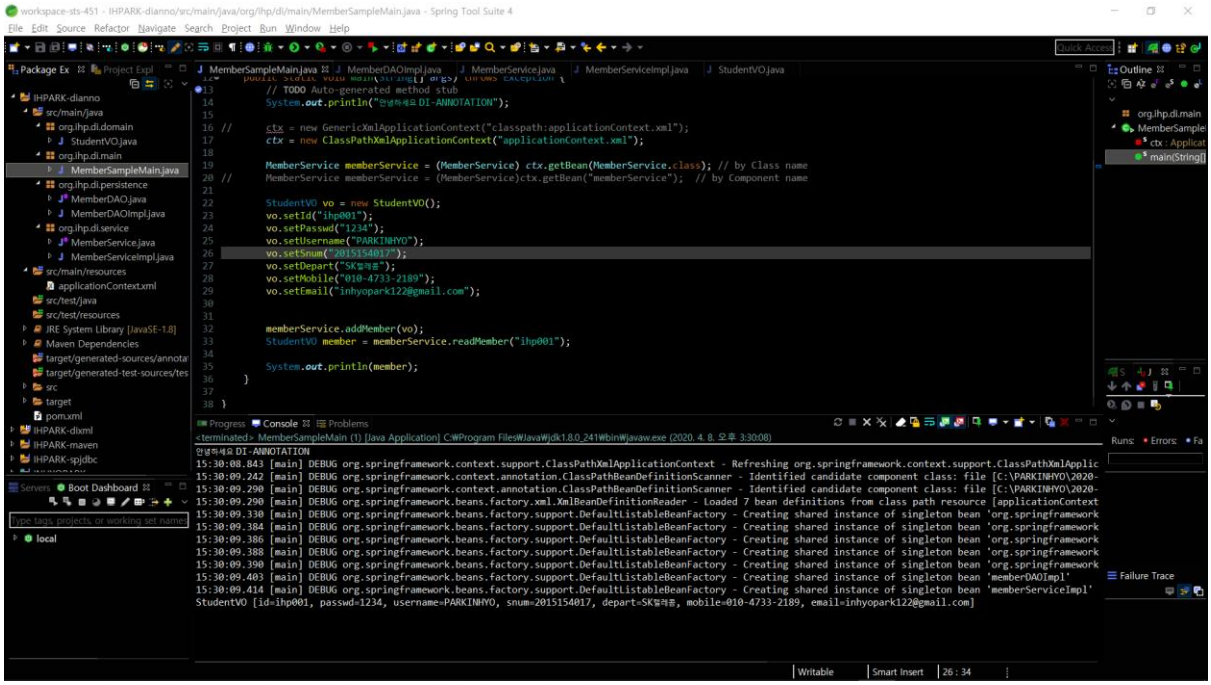
2) - ④ 인터페이스 구현

초기화 작업 : InitializeBean 인터페이스의 agterPropertiesSet 메소드를 구현

전처리 작업 : DisposableBean 인터페이스의 destroy 메소드를 구현

3) di-annotation.zip과 db-springjdbc.zip을 참고하여 실습하고 본인이 생성한 2개의 프로젝트 결과 출력 페이지를 PringScreen하여 첨부하시오.

3) - ① IHPARK-dianno



3) - ② IHPARK-spjdbc