[ 03 ] DI(Dependency Injection) 자동 설정과 빈 생명주기와 범위

1. XML 파일을 이용한 DI 설정방법 ; XML파일을 이용한 DI설정 방법은 그 동안 우리가 살펴본 방식 입니다. 이미 학습한 기본적인 사항들과 추가적인 새로운 사항들에 대해서 알아봅니다.

**@Autowired 어노테이션을 이용한 객체간 의존 자동 연결**

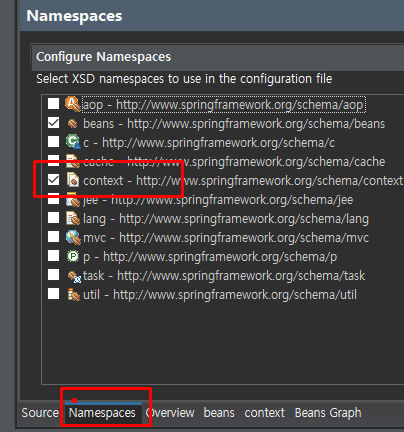
프로젝트의 규모가 조금만 커져도 한 개의 어플리케이션에서 생성하는 스프링 빈 객체는 수백개 이상으로 증가하게 되는데 이 경우 스프링 빈 간의 의존관계를 xml 설정이나 자바 기반 설정을 관리하는데 시간을 뺏길 수 있다. 또는 특정 타입의 빈 객체가 한 개밖에 존재하지 않는 경우가 많아서 의존 객체가 너무 뻔할 때가 있다. 만약 일일이 의존관계를 설정할 필요없이 자동으로 프로퍼티나 생성자 파라미터 값으로 동일 타입의 빈 객체를 전달해 주는 기능이 있다면 설정 코드가 많이 줄어들 것이다.

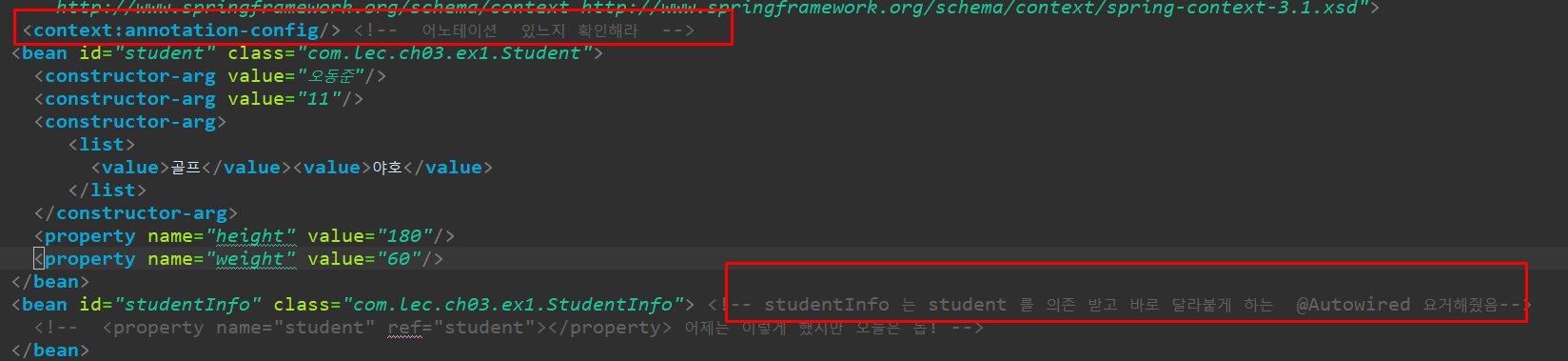
Ch03\_diLife

Spring MVC Project 로 웹프로젝트용

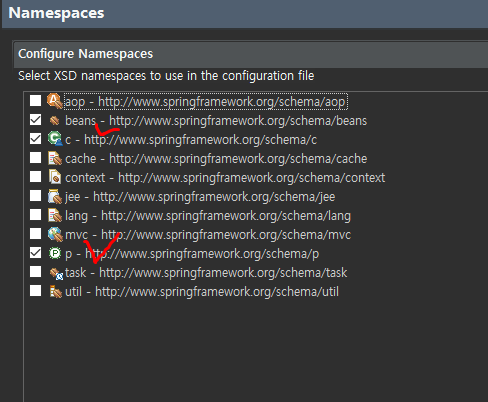
**Ex1 Student.java**

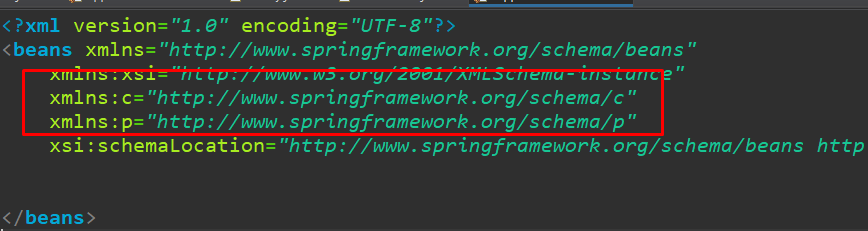
<!-- studentInfo 는 student 를 의존 받고 바로 달라붙게 하는 @Autowired 요거해줬음-->





*ex1.Family.java*

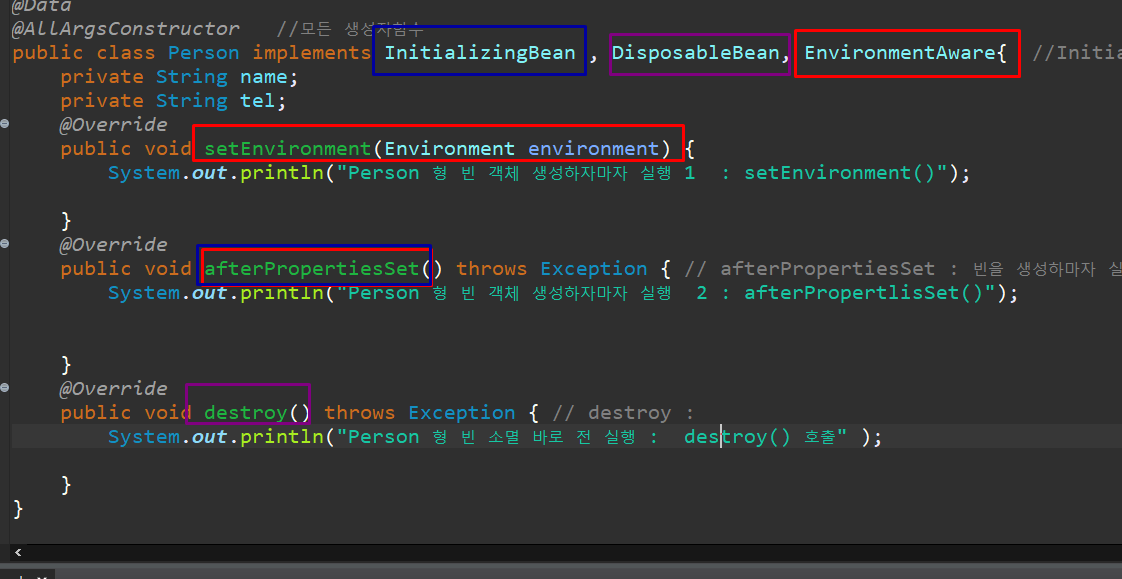




1. 스프링 컨테이너 생명 주기
   * 스프링 컨테이너 생성 : GenericXmlApplicationContext ctx = new GenericXmlApplicationContext();
   * 스프링 컨테이너 설정 : ctx.load(“classpath:applicationCTX.xml”); ctx.refresh();
   * 스프링 컨테이너 사용 : Student st = ctx.getBean(“student”,Student.class); st.getName();
   * 스프링 컨테이너 소멸(자원해제) : ctx.close();

Ex2.Person , OtherPerson

* Implements InitializingBean, DisposableBean – 인터페이스를 이용한 생명주기 관리 방법

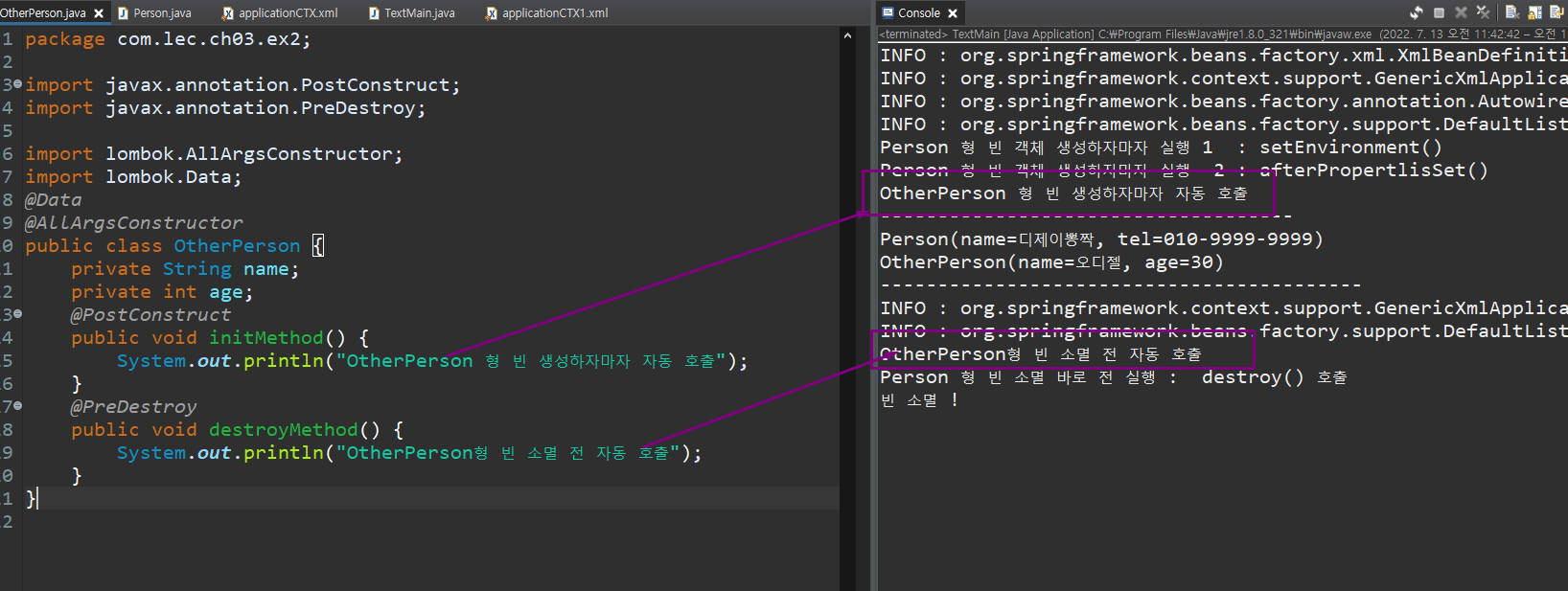


InitializingBean : afterPropertiesSet

DisposableBean : destroy

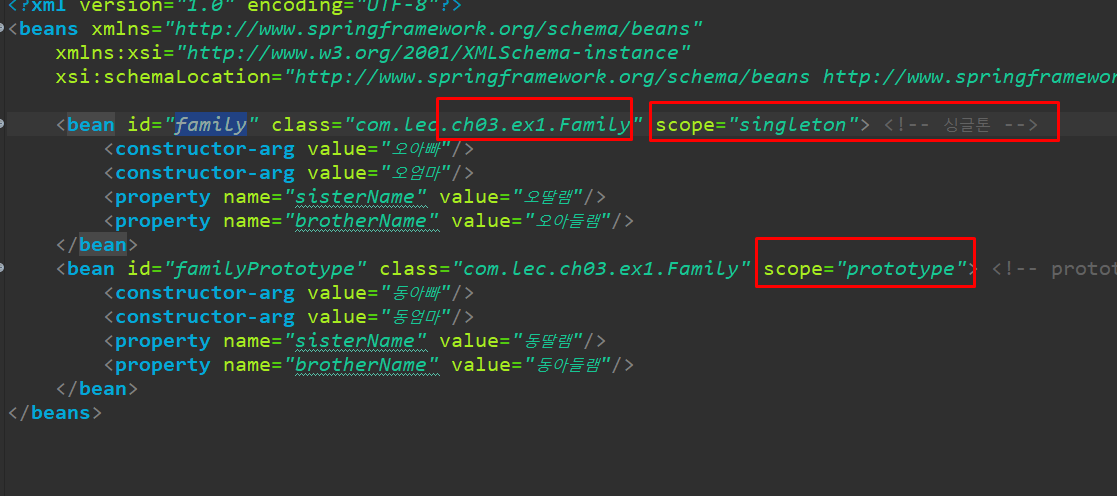
EnvironmentAware : setEnvironment

-OtherPerson.java

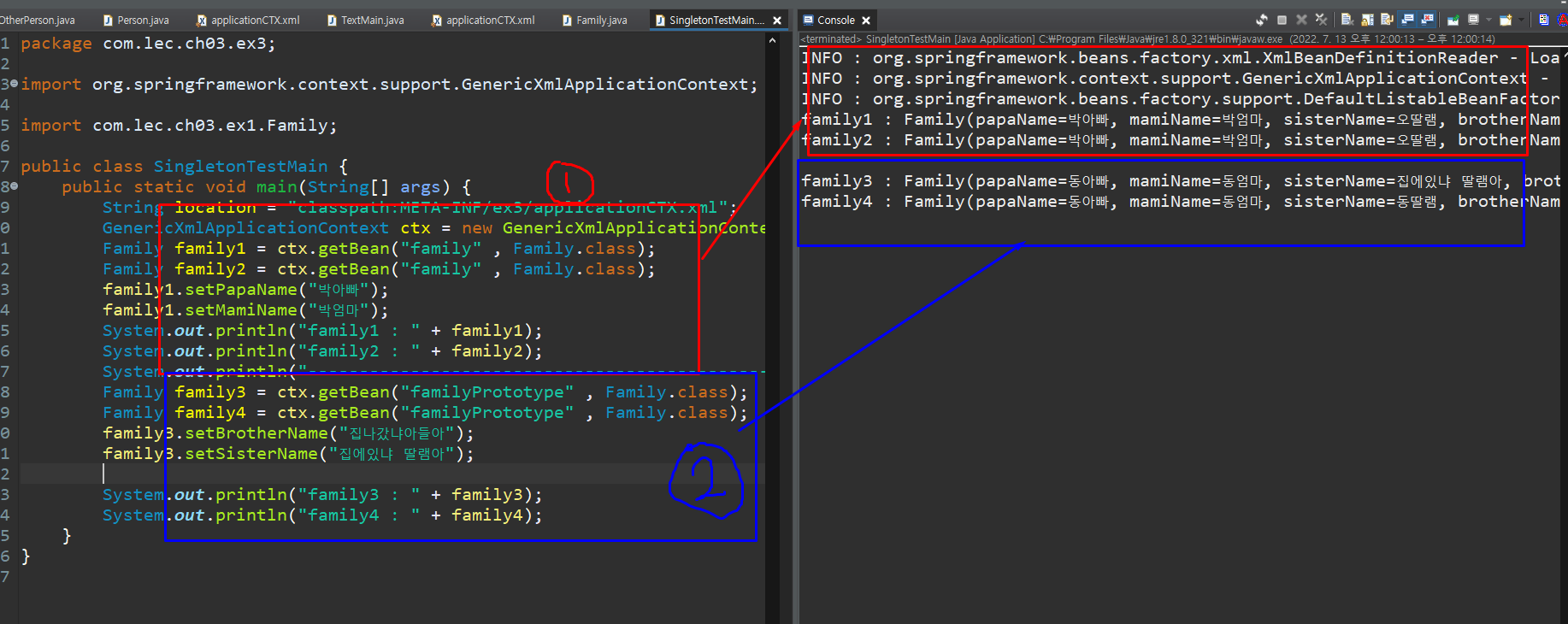


Ex3 SingletonTestMain.java 요약

* Ex3/ applicationCtx.xml



위에는 싱글톤 아래는 proto type 밑에서 실행 확인 해보자



* + - 1. 싱글톤이여서 하나만 바꿔도 전체 다 바뀐다
      2. Prototype 이여서 부분으로 바꿀수 있다 .