2024-02-21

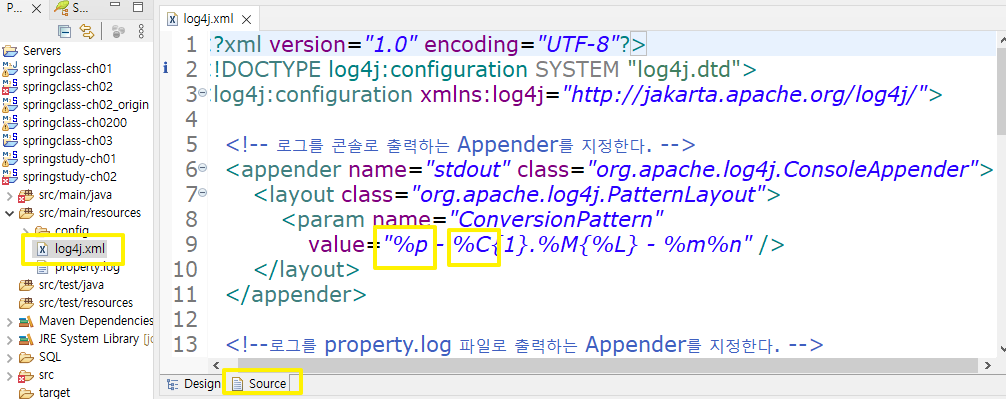
**P49 bean의 생명주기(Life Cycle) (~60p)**

Aop가 무엇인지 설명할 것(~75p, 여기까지 나갈 것)

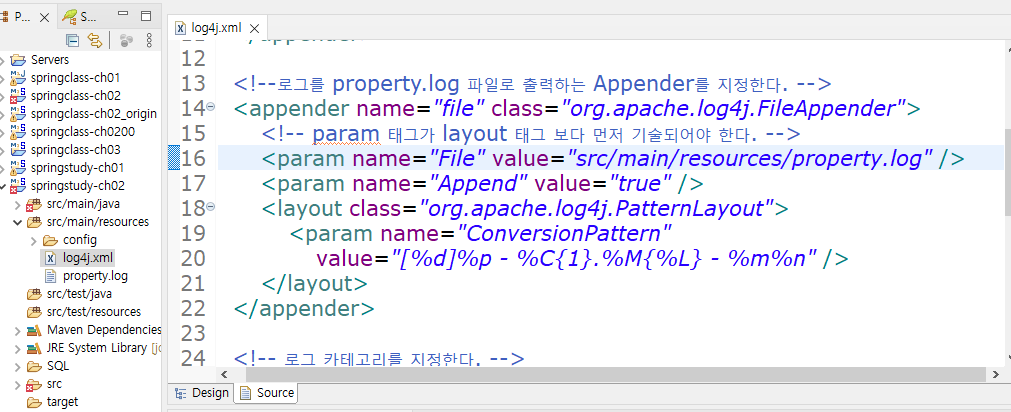
내일은 MVC할 것

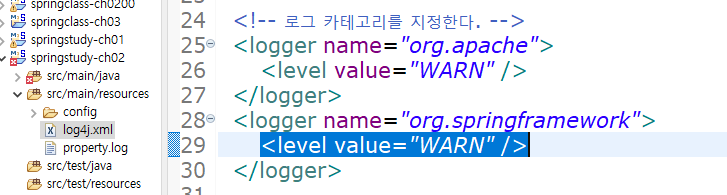
그림 2-4 는 중요하지 않음, 한번 읽어보기

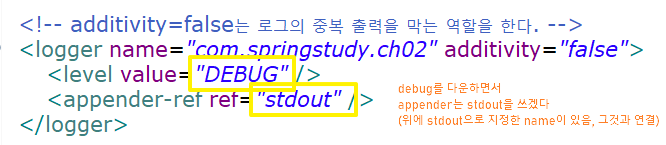
* Log4j의 구조 -> appender, logger, layout
* Log4j의 로깅 레벨 : 6단계

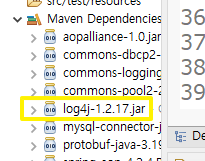
교안 계속 설명함  


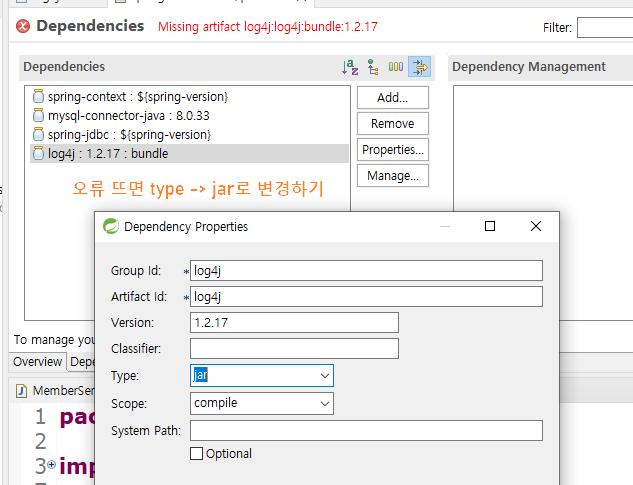
* Log4j의 로깅 환경 설정 방법

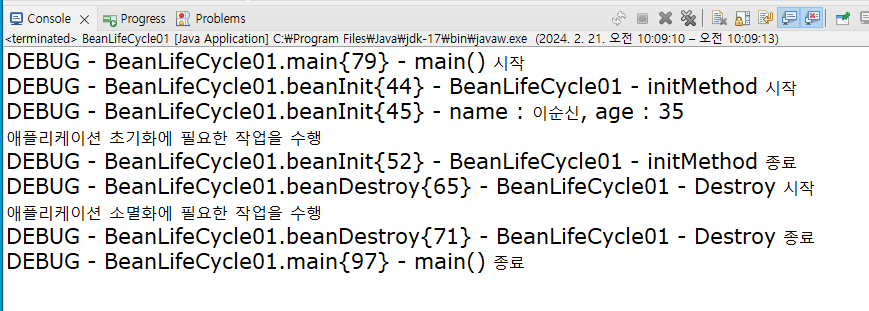




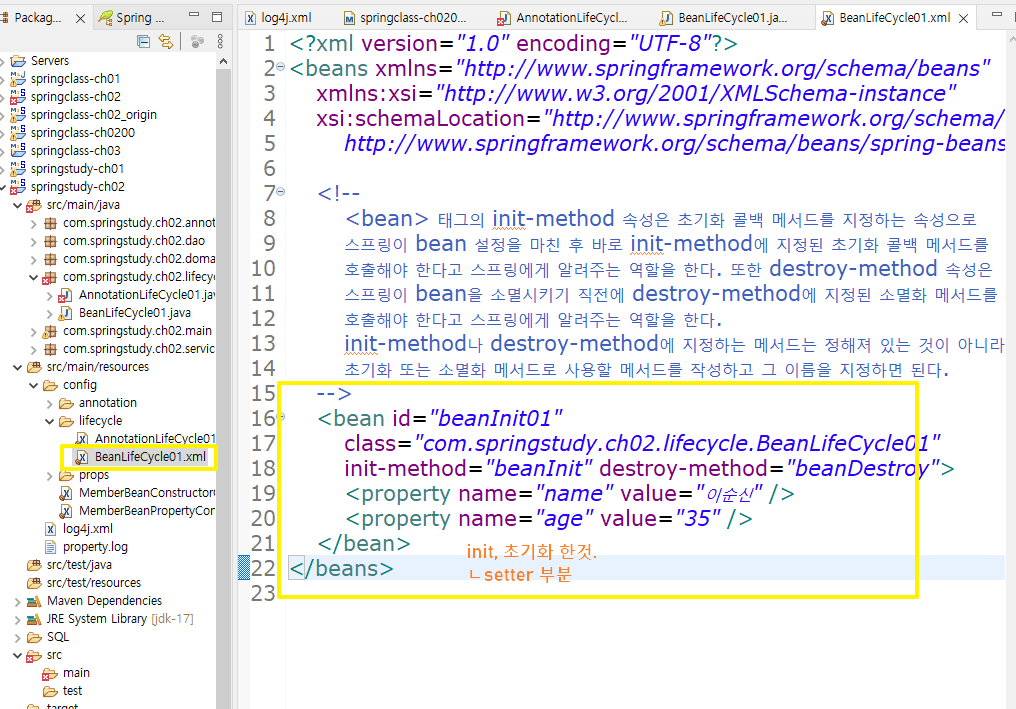




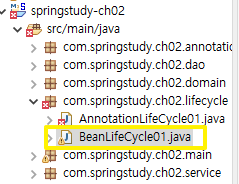


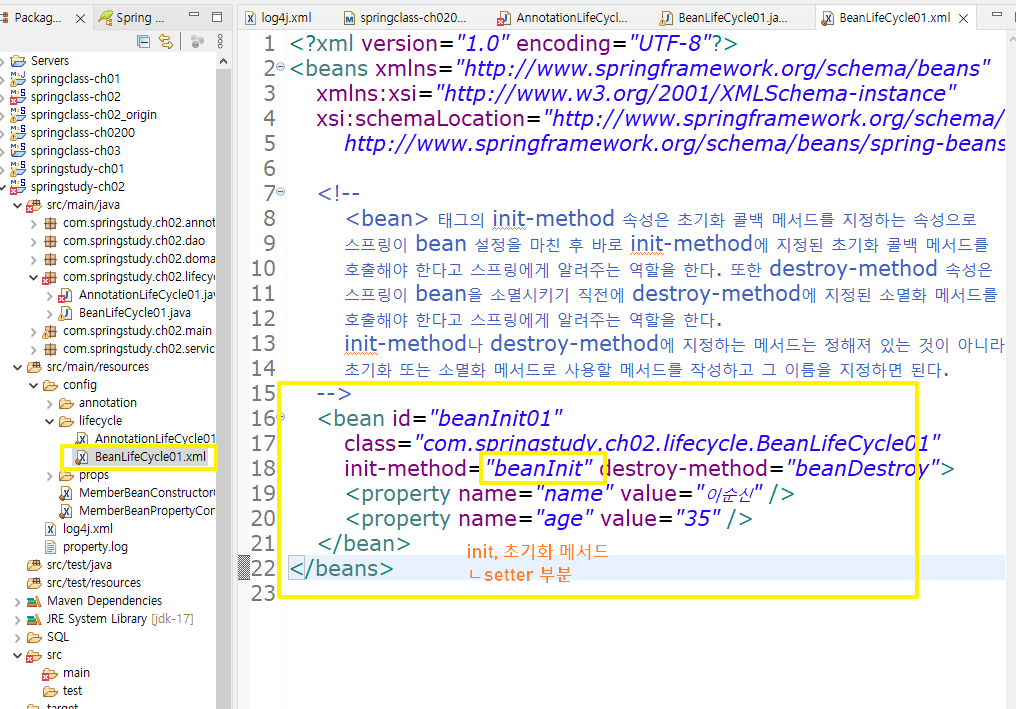
  
패턴 레이아웃 포맷을 참고하면 어떤 것을 출력하라는 것인지 알 수 있음(아래와 같이 출력됨)  


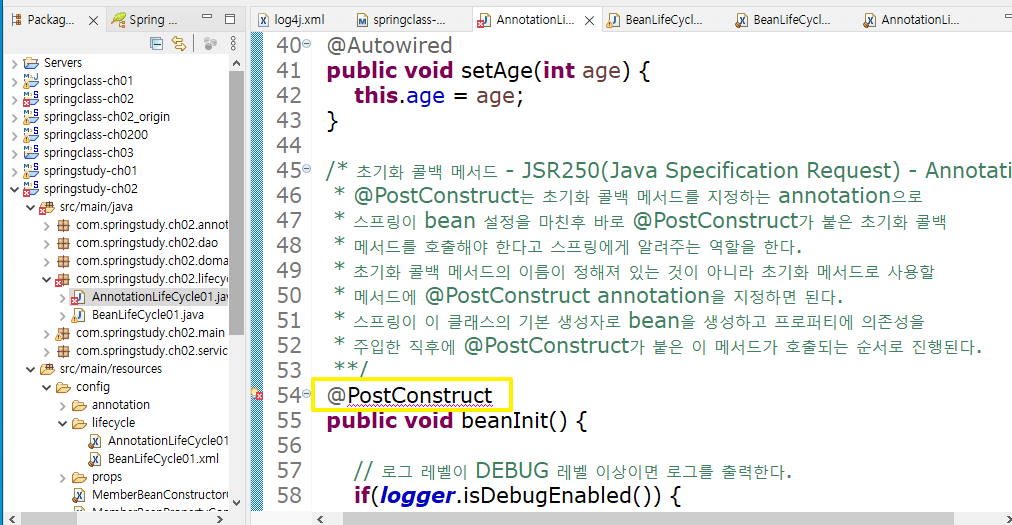
* 로깅처리를 위한 클래스 멤버설정과 메서드



* 초기화, 소멸화 작업이 잘 되는 것을 확인할 수 있음 (이 파일에서 코드 실행)



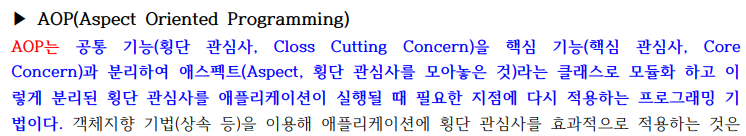




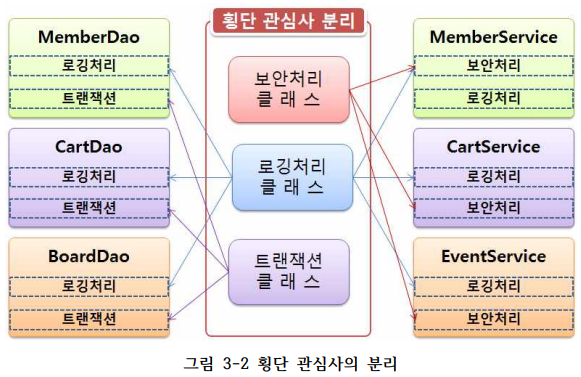
**P62 Spring AOP**

스프링 AOP란?  
애플리케이션 전반에 걸쳐 산재되어 있는 공통기능 분리

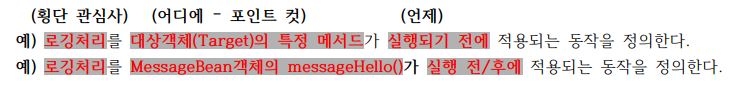
* 횡단관심사 > 로깅을 사용



로깅처리 따로 해줬어야 했는데, 공통기능을 가지고 와서 횡단관심사로 분리한 것

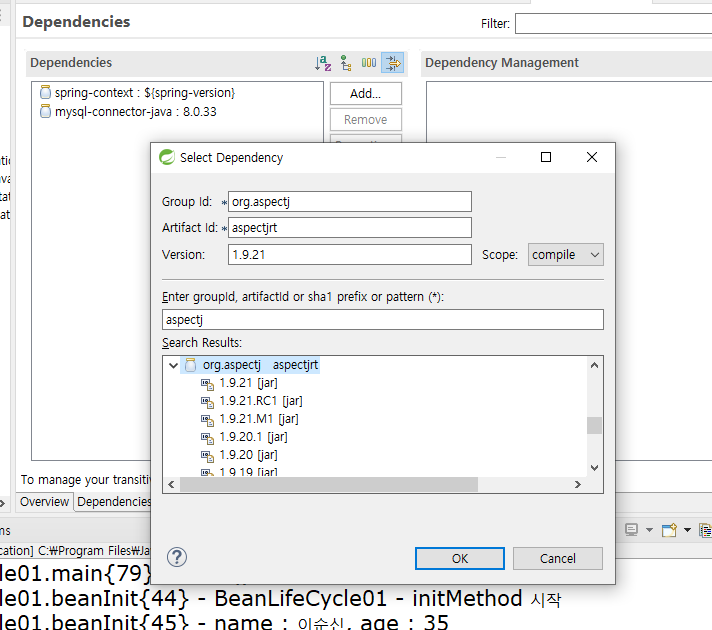


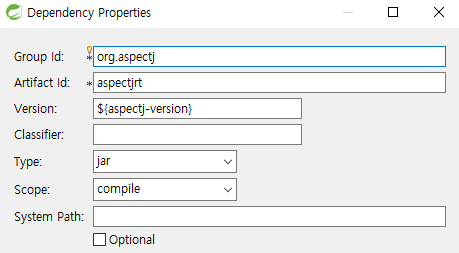
* 스프링 AOP 용어

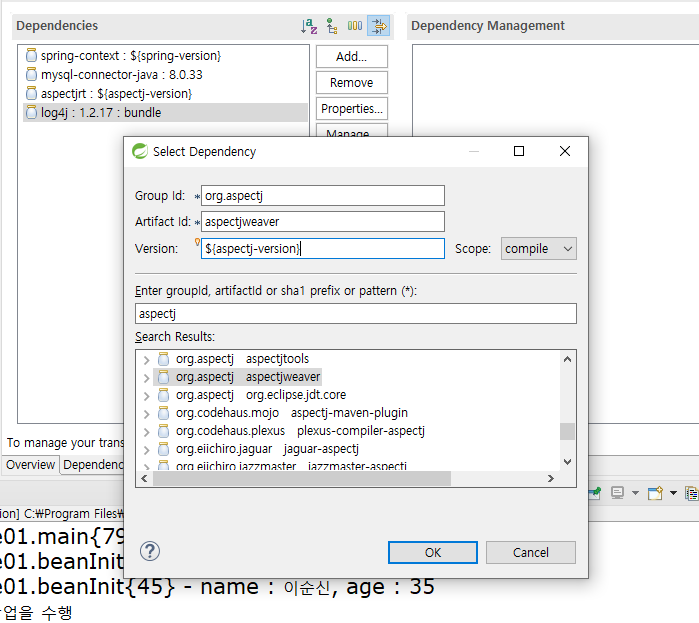
1. 어드바이스(Advice) : 공통기능 로그로 국한해서 생각하면되지만, 로그만 있는 건 아님
2. 조인포인트(JoinPoint) : 어드바이스를 적용할 수 있는 곳
3. 포인트 컷(pointcut) : 어드바이스를 어느 지점에 적용할지 정의한 것
4. 애스펙트(Aspect) : 한 개 또는 다수의 어드바이스와 포인트 컷의 조합  
   
5. 위빙(Weaving)
6. 대상 객체(Target)

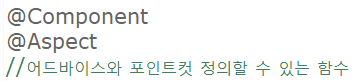
* 스프링 AOP 구현하기

파일명 : pringclass-ch03  
pom.xml에서 dependencies 설정해주기  
-> spring-context, aspectjrt, log4j, aspectjweaver (mysql은 내가 추가해줌 ㅋㅋ)

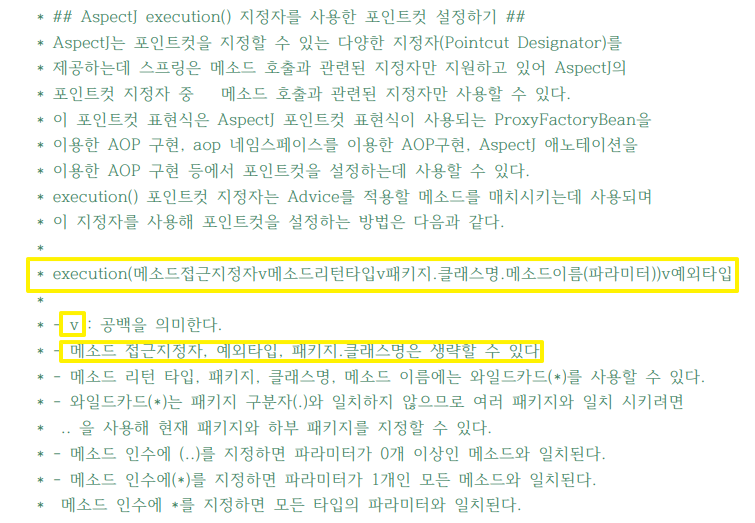
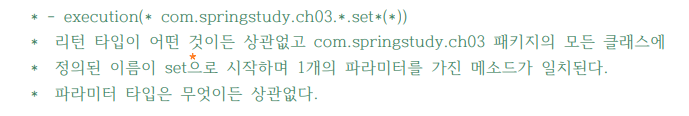
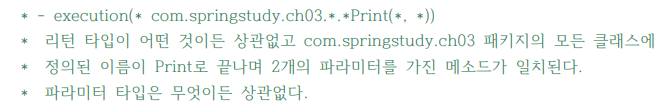




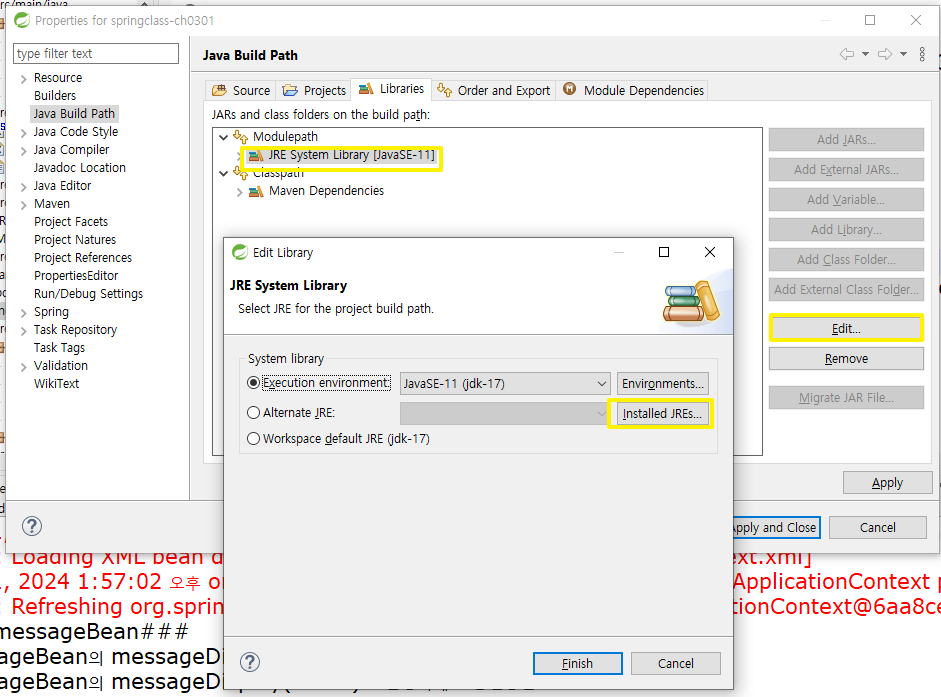
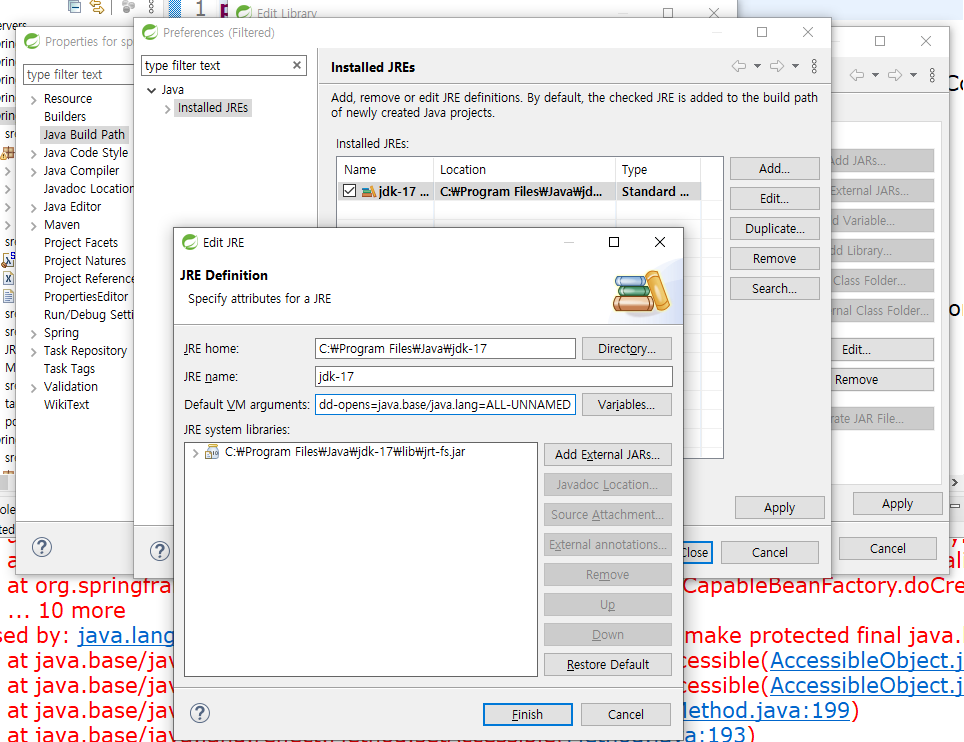
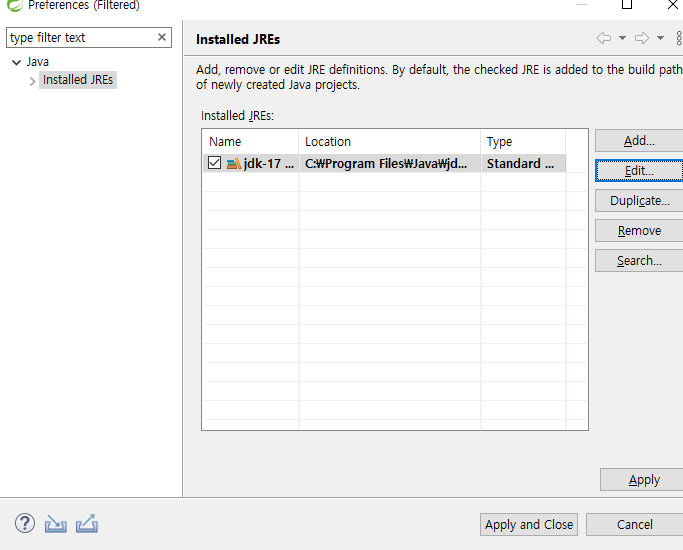




P68 설명  
- 그냥 다 읽어내려가면 됨

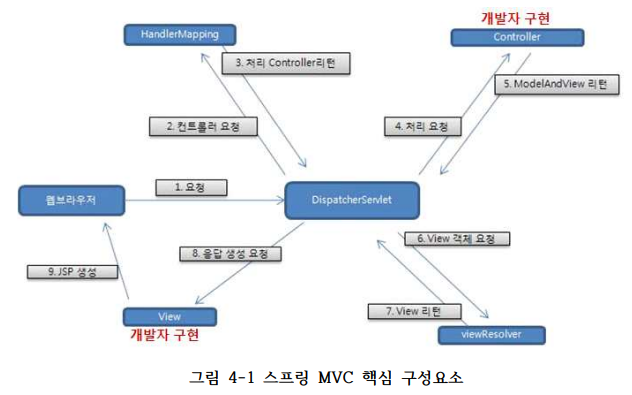
지정할 수 있음  
  
  


오후에 주신 파일에 자바버전이 맞지 않아서 오류 발생  
maven project -> springclass-ch03 오른쪽 마우스 > buildPath > configure build path

  
아래부분을 Default VM arguments 에 붙여넣기  
**--add-opens=java.base/java.lang=ALL-UNNAMED**  


**(p76) 4. SpringWebMVC**

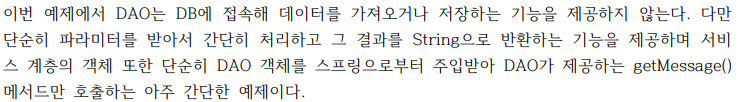
* Springstudy-ch0401

  
머리속에 외우기

1. dispatcherSErvlet에 요청을 받아서 분석  
   결국은 자동으로 등록이 됨

개발자구현 -> service, Model, 응답을 해줌

View

예제!  
DB 사용하는 예제가 아님, 계층구조 확인할 것.  


* 예제 하다가 문제가 있어서 내일 MVC 모델 진행하기로 함

Xml 어떻게 연결되는 건지 무슨역할, “이순신 왜 필요한건지”

**AspectContext.xml** : controller는 아니지만 그런 역할을 하는 것으로 bean을 등록해줌   
? c:\_0 = “이순신” 무슨 의미인지 질문

**MessageBeanAspectJAnnotation.java** : Bean을 선언해줌 함수를 선언하면 출력하도록 구성  
 오버라이딩해서 빈 값일 때, String일 때 출력하는 값을 다르게 정의해줌

**AspectMain.java** : 실행을 이 파일에서 해 줌, xml파일을 ac로 인스턴스 만들어주고,  
xml파일에서 만들어 놓은 bean을 get으로 얻어온다.   
그것은 MessageBeanAspectAnnotation.java에서 component해준 messageBean을 사용해서 만들어준다.

Memberservice

하나의 파일중에서

Annotation

Dao

Domain

Main

Service

Controller 가 model의 정보를 받아와서, view로 연결해줘서 view의 화면이