**11/28 김민식 OJT Day 19**

**Spring FrameWork 심화 교육 Day 01**

**1강. Framework 개념**

[Spring Framework Basic 1강 Framework 개념 | T아카데미 - YouTube](https://www.youtube.com/watch?v=j5WlADp5f_o&list=PL9mhQYIlKEhfYqQ-UkO2pe2suSx9IoFT2)

1 SW 재사용 방안들

* 1. 복사(Copy) & 붙이기(Paste)
* 가장 기본적인 재사용 방안
  1. 메서드 호출
  2. 클래스 재사용(상속)
  3. AOP(Aspect Oriented Programming)
* 관심의 분리

2 디자인 패턴과 프레임 워크의 관련성

* 디자인 패턴
  + 프로그램 개발에서 자주 나타나는 과제를 해결하기 위한 방법
    - 사용 하는 이유 : 요구사항이 수시로 변경시 Source Code 변경을 최소화
    - 인수 인계 편리
* 프레임워크
  + 비기능적인 요소들을 초기 개발 단계마다 구현해야 하는 불합리함을 극복
  + 반복되는 문제를 해결하기 위한 특화된 Solution을 제공
* 디자인패턴과 프레임워크의 관련성
  + 디자인 패턴은 프레임워크의 핵심적인 특징

3 프레임워크의 구성요소와 종류

3-1. IoC(Inversion of Control)

* 제어의 역전
  + 인스턴스 생명주기 관리를 개발자가 아닌 컨테이너가 대신 해준다.
  + 일반적인 프로그램 흐름과 반대로 동작

3-2. 클래스 라이브러리 (Class Library)

* 유저코드가 라이브러리를 호출
* 제어를 개발자의 코드가 관장
  + But. 프레임워크는 프레임워크가 코드를 관장

3-3. 디자인 패턴

* 디자인 패턴 + 라이브러리 = 프레임워크

3-4. 프레임워크 종류

* 웹(MVC) : Spring MVC
* OR(Object-Relational) 매핑 : MyBatis
* AOP(Aspect Oriented Programming) : Spring AOP
* DI(Dipendency Injection) : Spring DI
* Build와 Library 관리 : Maven
* 단위 테스트 : jUnit
* JavaScript : jQuery

**3강. 프레임워크 개요**

[Spring Framework Basic 3강 Spring 프레임워크 개요 | T아카데미 - YouTube](https://www.youtube.com/watch?v=yoQwdSEGoK0&list=PL9mhQYIlKEhfYqQ-UkO2pe2suSx9IoFT2&index=3)

1 Spring Framework의 정의

* Java 엔터프라이즈 개발을 편하게 해주는 오픈소스 경량급 애플리케이션 프레임워크
  + 경량급 : 단순한 웹컨테이너에서도 엔터프라이즈 개발의 고급 기술을 대부분 사용 가능
  + 애플리케이션 프레임워크: 특정 계층이나 기술, 업무 분야에 국한되지 않고 범용적인 프레임워크
  + 엔터프라이즈 개발 용이 : low Level(보안, 인증, Tx 처리)d에 많이 신경 쓰지 않으면서 개발에 전념
  + 오픈소스 : 오픈소스의 장점과 제품의 단점과 한계를 극복

2 Spring Framework의 전략

* Portable Service Abstraction (서비스 추상화)
  + 기술적인 복잡함은 추상화를 통해 Low Level의 기술 구현 부분과 기술을 사용하는 인터페이스로 분리
* DI (의존관계주입)
  + 객체 지향에 출실한 설계가 가능
* AOP (관점 지향 프로그래밍)
* POJO(Plain Old Java Object)
  + JVM만 있으면 돌아가는 코드

3 Spring Framework의 특징

* 컨테이너 역할
* DI 지원
  + 설정파일이나 어노테이션으로 설정
* AOP 지원
  + 로깅, 보안과 같은 공통적인 부분 분리
* POJO 지원
* 트랜잭션 처리를 위한 일관된 방법을 지원
* 영속성과 관련된 다양한 API 지원

4 Spring Framework 기능요소

* Core 컨테이너 : 기본기능을 제공, 이 안에 BeanFactory는 Spring의 기본 컨테이너
* AOP : 업무로직과 공통로직 분리
* ORM : MyBatis 등 사용가능
* DAO : 예외처리 간편화, 트랜잭션 관리
* Web : 기웹애플리케이션에 필요한 기본 기능 제공
* Context : BeanFactory의 개념을 확장 / 국제화 메시지 등 지원
* webMVC : 애플리케이션 로직과 웹 애플리케이션로직을 분리

**4강. 프로젝트 시작하기**

[Spring Framework Basic 4강 Spring 프로젝트 시작하기 | T아카데미 - YouTube](https://www.youtube.com/watch?v=F8XsFLFzQuo&list=PL9mhQYIlKEhfYqQ-UkO2pe2suSx9IoFT2&index=4)

1 STS 소개 및 제공하는 기능

* 1. STS(SpringSource Tool Suite) 소개
* SpringSource가 직접 만들어 제공하는 이클립스의 확장판
* Spring 개발에 최적화 되도록 만들어진 IDE
  1. STS가 제공하는 기능
* Bean 클래스 이름 자동완성
* 설정파일 생성 위저드
* Bean 의존관걔 그래프
* AOP 적용 대상 표시

2 Maven과 Library 관리

* 1. Maven이란?
* 라이브러리 관리 – 빌드 툴
* 사용하는 이유
  + 편리한 Dependent Library 관리
  + 여러 프로젝트에서 프로젝트 정보나 jar파일을 공유하기 쉬움
  + 빌드 프로세스를 일관되게 설정 가능
* Pom.xml
  + Project Object Model 정보를 담고 있다

**5강. IoC 컨테이너와 DI(Dependency Injection)**

[Spring Framework Basic 5강 Ioc컨테이너와 DI(Dependency Injection) | T아카데미 - YouTube](https://www.youtube.com/watch?v=GKoCibDM6Ns&list=PL9mhQYIlKEhfYqQ-UkO2pe2suSx9IoFT2&index=5)

1 IoC(Inversion of Control)

* 제어권의 역전, 객체의 생성, 생명주기의 관리까지 모든 객체에 대한 제어권이 바뀌었다는 것을 의미
* 컴포넌트 의존관계 설정, 설정 및 생명주기를 해결하기 위한 디자인 패턴
* IoC 컨테이너
  + 객체에 대한 생성 및 생명주기를 관리, 의존성을 관리
  + POJO의 생성, 초기화, 서비스, 소멸에 대한 권한
* IoC의 분류
  + DL : Dependency Lookup - 의존성 검색
  + DI : Dependency Injection – 의존성 주입

2 DI

* DI의 개념
  + 의존관계를 빈 설정 정보를 바탕으로 컨테이너가 자동으로 연결
  + 컨테이너가 흐름의 주체가 되어 애플리케이션 코드에 의존관계를 주입
* DI의 유행
  + Setter Injection : Setter 메서드를 이용한 의존성 삽입
  + Constructor Injection : 생성자를 이용한 의존성 삽입
  + Method Injection : 일반 메서드를 이용한 의존성 삽입

2 Spring DI 컨테이너

* 개념
  + Spring DI 컨테이너가 관리하는 객체를 Bean, 해서 컨테이너를 빈 팩토리
* BeanFactory와 ApplicationContext
  + BeanFactory
    - Bean을 등록, 생성, 조회, 반환 관리
    - GetBean() 메서드가 정의되어 있음
  + ApplicationContext
    - Spring의 각종 부가 서비스를 추가로 제공

**6강. DI 애플리케이션 작성(1)**

[Spring Framework Basic 6강 DI 애플리케이션 작성(1) | T아카데미 - YouTube](https://www.youtube.com/watch?v=-Y9811mmTiQ&list=PL9mhQYIlKEhfYqQ-UkO2pe2suSx9IoFT2&index=6)

* 실습 프로젝트 설정( 빈 생성)
  + Maven 파일로 전환하고 add Spring
  + Java version 때문에 오류
  + Maven dependency 때문에 오류

**7강. DI 애플리케이션 작성(2)**

[Spring Framework Basic 7강 DI 애플리케이션 작성(2) | T아카데미 - YouTube](https://www.youtube.com/watch?v=dJavXu6DB9s&list=PL9mhQYIlKEhfYqQ-UkO2pe2suSx9IoFT2&index=7)

1 jUnit의 개요와 특징

* assert메소드로 테스트 케이스의 수행 결과를 판별한다.
* jUnit4부터는 테스트를 지원하는 어노테이션을 제공

1. jUnit을 사용한 DI테스트 클래스

* 테스트를 지원하는 어노테이션
  + @Test : 선언된 메서드는 테스트를 수행하는 메소드가 된다.
  + @ignore : 테스트를 실행하지 않게 한다.
  + @Before : @test 전에 실행
  + @After : @test 후 실행
  + @BeforeClass : @Test 메소드 보다 먼저 한번만 수행
  + @AfterClass : @Test 메소드 보다 후에 한번만 수행
* 테스트 결과를 확인하는 단정(assert) 메서드

1. Spring-Test를 사용한 DI 테스트 클래스

* 테스트를 지원하는 어노테이션
  + @RunWith : jUnit 프레임워크의 테스트 실행방법을 확장할 때 사용하는 어노테이션
  + @ContextConfiguration : 스프링 빈 설정 파일의 위치를 지정할 때 사용
  + @Autowired : 스프링 DI에서 사용, 빈을 자동으로 매핑

질문사항

1. 어노테이션 자동완성 x