



# JAVA 웹 개발자 양성과정

## Spring: Spring Core

### 6강 - 의존관계 자동주입

By SoonGu Hong



# JAVA 웹 개발자 양성과정

## Spring: Spring Core

### 1. 자동 주입

## 다양한 의존관계 주입 방법

의존관계 주입은 크게 4가지 방법이 있다.

- 생성자 주입
- 수정자 주입(setter 주입)
- 필드 주입
- 일반 메서드 주입

## 생성자 주입

- 이름 그대로 생성자를 통해서 의존 관계를 주입 받는 방법이다.
- 지금까지 우리가 진행했던 방법이 바로 생성자 주입이다.
- 특징
  - 생성자 호출시점에 딱 1번만 호출되는 것이 보장된다.
  - **불변, 필수** 의존관계에 사용

```
@Autowired
public OrderServiceImpl(MemberRepository memberRepository, DiscountPolicy
discountPolicy) {
    this.memberRepository = memberRepository;
    this.discountPolicy = discountPolicy;
}
```

## 수정자 주입(setter 주입)

- setter라 불리는 필드의 값을 변경하는 수정자 메서드를 통해서 의존관계를 주입하는 방법이다.
- 특징
  - 선택, 변경 가능성이 있는 의존관계에 사용
  - 자바빈 프로퍼티 규약의 수정자 메서드 방식을 사용하는 방법이다.

```
@Autowired
```

```
public void setMemberRepository(MemberRepository memberRepository) {  
    this.memberRepository = memberRepository;  
}
```

```
@Autowired
```

```
public void setDiscountPolicy(DiscountPolicy discountPolicy) {  
    this.discountPolicy = discountPolicy;  
}
```

## 필드 주입

- 이름 그대로 필드에 바로 주입하는 방법이다.
- 특징
  - 코드가 간결해서 많은 개발자들을 유혹하지만 외부에서 변경이 불가능해서 테스트 하기 힘들다는 치명적인 단점이 있다.
  - DI 프레임워크가 없으면 아무것도 할 수 없다.
  - 사용하지 말자!
    - 애플리케이션의 실제 코드와 관계 없는 테스트 코드
    - 스프링 설정을 목적으로 하는 @Configuration 같은 곳에서만 특별한 용도로 사용

```
@Autowired
```

```
private MemberRepository memberRepository;
```

```
@Autowired
```

```
private DiscountPolicy discountPolicy;
```

## 생성자 주입을 선택해라!

과거에는 수정자 주입과 필드 주입을 많이 사용했지만, 최근에는 스프링을 포함한 DI 프레임워크 대부분이 생성자 주입을 권장한다. 그 이유는 다음과 같다.

### 불변

- 대부분의 의존관계 주입은 한번 일어나면 애플리케이션 종료시점까지 의존관계를 변경할 일이 없다. 오히려 대부분의 의존관계는 애플리케이션 종료 전까지 변하면 안된다.(불변해야 한다.)
- 수정자 주입을 사용하면, setXxx 메서드를 public으로 열어두어야 한다.
- 누군가 실수로 변경할 수 도 있고, 변경하면 안되는 메서드를 열어두는 것은 좋은 설계 방법이 아니다.
- 생성자 주입은 객체를 생성할 때 딱 1번만 호출되므로 이후에 호출되는 일이 없다. 따라서 불변하게 설계할 수 있다.