6강 DI ( Dependency Injection)

DI를 이용한 프로그래밍 방법과 스프링에서의 DI에 대해서 학습합니다.

1. 의존주입이라는 방법은 객체지향에서 쓰는 방법 (스프링에서만 사용되는 방법은 아님)
2. 스프링에서 DI란?

>> 배터리 분리형 이라고 보면된다.

>> 배터리 일체형 같은경우 배터리가 떨어지면 배터리 교체가 힘들다.

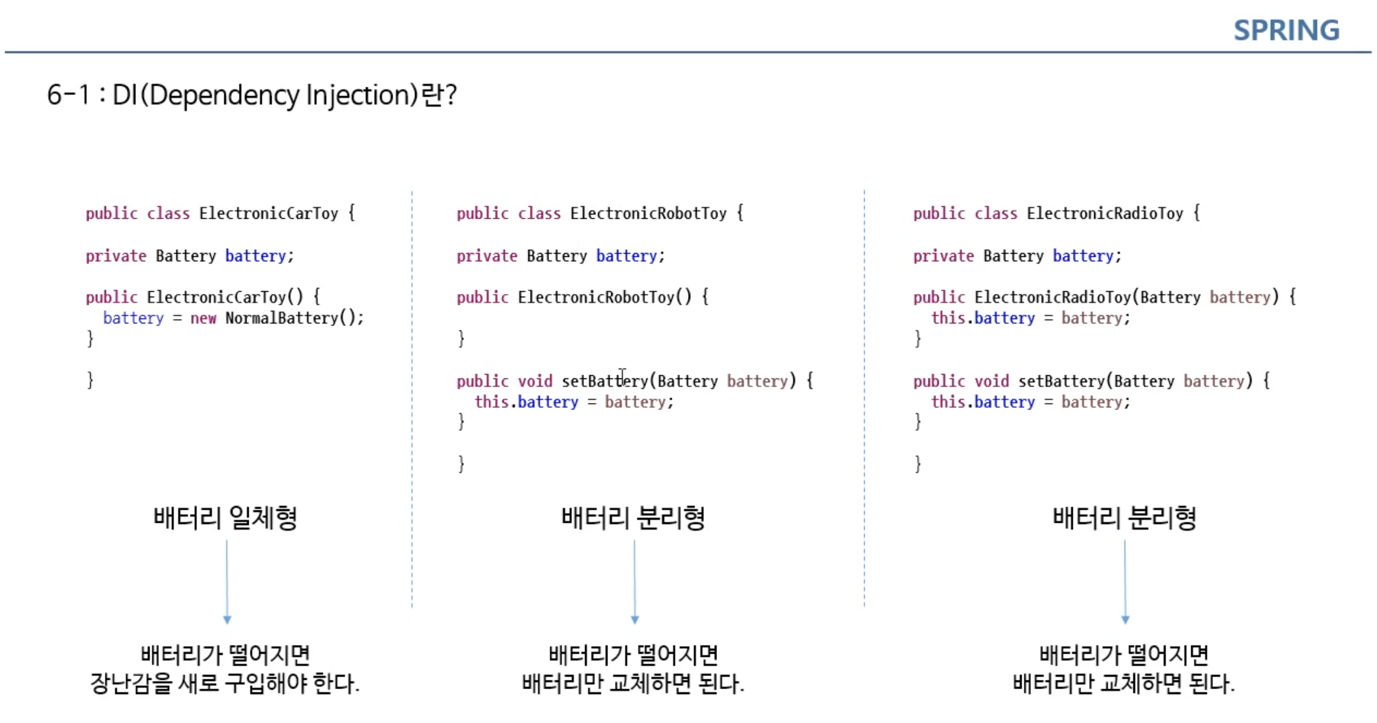
하지만 배터리 분리형은 배터리가 떨어지면 배터리만 교체하면 된다

>> 즉, 약한 결합을 실현시켜줌 !

>> 프로그래밍적으로 생각한다면 , 수많은 객체를 통해 어떤 프로그램을 만들었다고 했을때

객체 하나하나 모두 강하게 엮어져 있다면 유지보수가 힘들어진다.

객체를 떼었다 붙였다 하는 것이 쉬워질수록 유지보수가 쉬워진다.



생성자 >> 만들때부터 배터리를 넣어서 만든다.

setter >> 원하는 배터리를 넣어라

생성자 + setter >> 만들때부터 배터리를 넣어주고 언제든지 교체가 가능하다.

세가지 경우 모두 주입이지만 구조적인 차이가 있다.

(세번째가 유연하고 견고)

1. 스프링에서의 의존주입

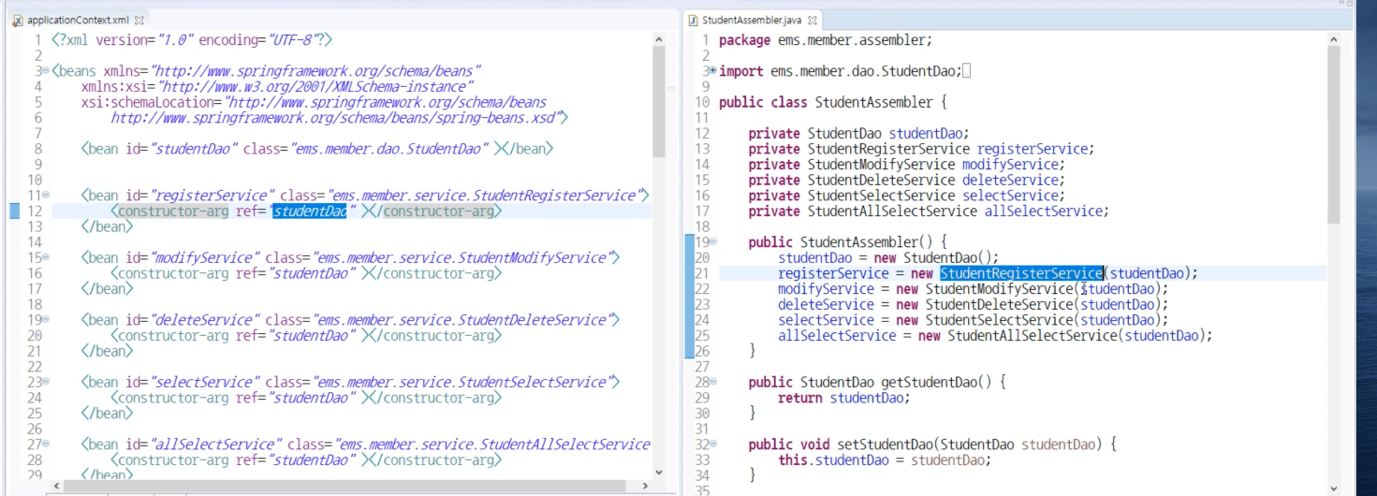
>> 스프링 컨테이너(applicationContext.xml) 에 bean객체를 만들어서 사용

>> 이 bean들사이에서 의존주입이 일어날수 있음

>> 즉, bean 안에 작은 bean이 들어가는 형태

>> 의존주입이 이루어져 있는 객체를 불러다 사용할 수 있다.

(예 : 배터리라는 객체가 들어가있는 로봇이라는 객체)



bean안에 bean을 넣을때 constructor-arg를 통해 넣을 수 있다. >>생성자로 넣는것과 같은 효과