

# JPA 05

자바 표준 ORM 표준 JPA 프로그래밍

김영한

2022.08.06

---

## 값 타입

### JPA의 데이터 타입 분류

- 엔티티 타입 : **@Entity**로 정의하는 객체 추적 가능 ( 값이 변해도 식별자로 추적 )
- 값 타입 **int, Integer, String** 처럼 단순히 값으로 사용하는 자바 기본타입 객체 추적 불가능 (식별자가 없고 값만 있기 때문에 )

### 값 타입 분류

#### 기본 값 타입

- 자바 기본 타입 (int, double)
- 래퍼 클래스 ( Integer, Long )
- String

생명주기를 엔티티의 의존 ( 회원을 삭제하면 나이 필드도 함께 삭제 )

값 타입은 공유하면 **X** ( 사이드 이펙트가 발생하면 안된다 )

기본 타입은 항상 값을 복사한다 -> 공유가 불가능

Integer , String 은 클래스 지만 공유가 불가능하다.

#### 임베디드 타입

- 내장 타입 ( 새로운 값 타입을 직접 정의할 수 있음 ) embedded type 여러 기본 타입을 모아서 만들어서 복합으로 만듦

- 
- 공통 주제의 속성으로 묶어서 공통 타입으로 만들어 클래스 타입으로 만들고 싶을 때 사용한다.
  - 예 ) 회원 : 이름, 나이, 근무 시작일, 근무 종료일 -> 이름, 나이, 근무 기간 이런식.

**@Embeddable** : 값 타입을 정의하는 곳에서 사용

**@Embedded** : 값 타입을 사용 하는 곳에서 사용

장점

재사용, 높은 응집도, 해당 값 타입만 사용하는 의미있는 메소드를 만들 수 있다, 값 타입을 소유한 엔티티에 생명주기가 같아진다.

**@AttributeOverride** : 속성 재정의 → 같은 속성을 두개 이상 사용할 때 사용

값 타입과 불변 객체

값 타입은 복잡한 객체를 조금이라도 단순화 하려고 만든 개념이니 단순하고 안전하게 사용해야 한다.

값 타입 공유 참조

임베디드 타입 값은 값 타입을 여러 엔티티에서 공유하면 위험 하다. 부작용 발생

값을 복사해서 사용해야한다. → 잘못 사용시에 막을 방법이 없다.

자바 기본 타입이 아니라 객체 타입이다 (즉, 객체 타입은 대입하면 값이 아닌 참조값이 복사된다 )

객체의 공유 참조는 피할 수 없다. → 불변 객체로 만든다.

불변 객체

생성 시점에만 값을 생성하고 변경이 불가능 하게 생성한다.

setter를 생성 하지 않는다.( private )

Integer , String, 래퍼클래스들 은 대표적인 불변 객체이다.

---

⇒ 변경하고 싶으면 값을 새로 만들어서 갈아준다. ( '값'은 부분 변경하는것이 어색한것이 맞다 )

## 값 타입의 비교

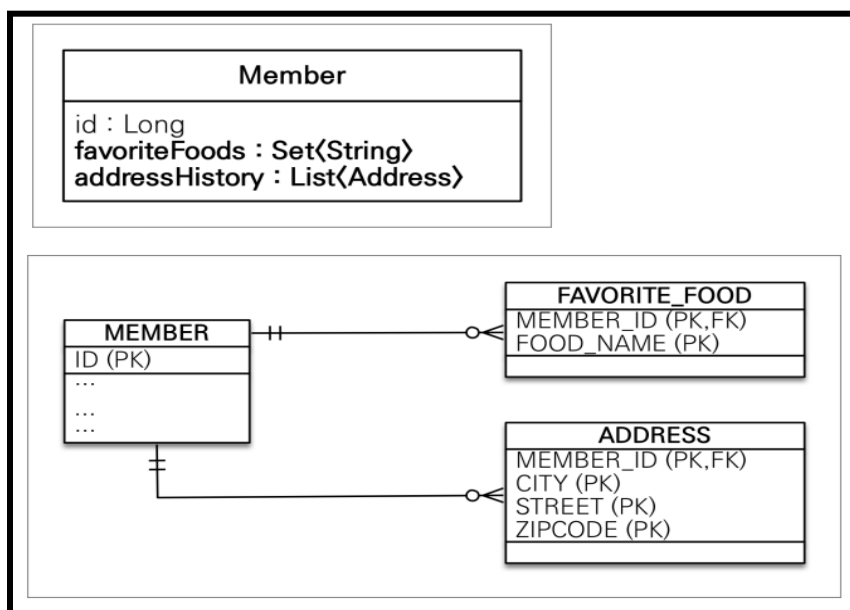
인스턴스가 달라도 그 안에 값이 같으면 같은 것으로 봐야 함.

즉, 클래스의 비교

동일성(**identity**) 비교 : 인스턴스의 참조 값을 비교 , == 비교

동등성(**equivalence**) 비교 : 인스턴스의 값을 비교 , equals() 비교

## 값 타입 컬렉션



이러한 경우 별도의 테이블로 뽑아 내야 한다.

- 값 타입을 하나 이상 저장할 때 사용
- @ElementCollection, @CollectionTable 사용해서 사용한다.
- 같은 테이블에 저장하는 것이 아닌 새로운 테이블을 생성하는것.

em.persist 하게 되면 CASCADE ALL 처럼 같이 persist 되고 라이프 사이클이 같이 돌게 된다.

정리하게 되면 [cascade + 고아 객체 제거 기능]을 필수로 가지고 있는것.

- 
- + 각 값 타입 컬렉션 들은 전부 지연로딩이다.

## 값 타입 컬렉션 수정

영속성 전이 처럼 컬렉션 값을 수정하면 테이블이 업데이트 된다.

-테이블에 존재하고 연관된 데이터를 전부 삭제되고 업데이트 된 컬렉션을 전부 삽입한다.

## 값 타입 컬렉션의 제약 사항

- 값 타입은 엔티티와 다르게 식별자 개념이 없다 → 변경 추적이 어렵다.
- 값 타입 컬렉션에 변경 사항이 발생하면 주인 엔티티와 관련된 모든 데이터를 삭제하고 ,  
값 타입 컬렉션에 있는 현재 값을 모두 다시 저장한다.
- 모든 컬럼을 묶어서 기본키로 구성해야한다 null입력X, 중복 저장 X  
  
↳ 사용 안하는걸 추천
- 그냥 일대다 엔티티 구조를 사용하자

정리 : 값 타입은 정말 값 타입이라 판단 될 때만 사용

참고: equals 를 재정의할때 getter를 사용하는게 안전 ( 프록시때 제대로 멤버접근이 안된다)

---

---

---