

## 10-2. 임베디드 타입

내가 만들어서 쓸 수 있는 타입!

만들었다고 해도 값 타입으로 취급된다.

클래스 두개를 새로 뽑아서 값 타입으로 취급하여 사용한다.

임베디드 타입은 그냥 엔티티에 속해 있는 값 타입일 뿐이다.

따라서 생명주기를 의존하고 있는 엔티티가 죽으면 애도 같이 죽는다.

`@Embeddable` : 정의 시 사용

`@Embedded` : 사용 시 사용

이 두 어노테이션으로 사용이 가능함.

`@Period.isWork()` 처럼 임베디드 타입만 쓸 수 있는 메소드를 사용가능.

→ 매우 객체지향적!

`@Embedded` 로 임베디드 타입 선언 시 기본 생성자는 필수!

기본 생성자를 갖고 있어도 다른 생성자를 가질 수 있다.

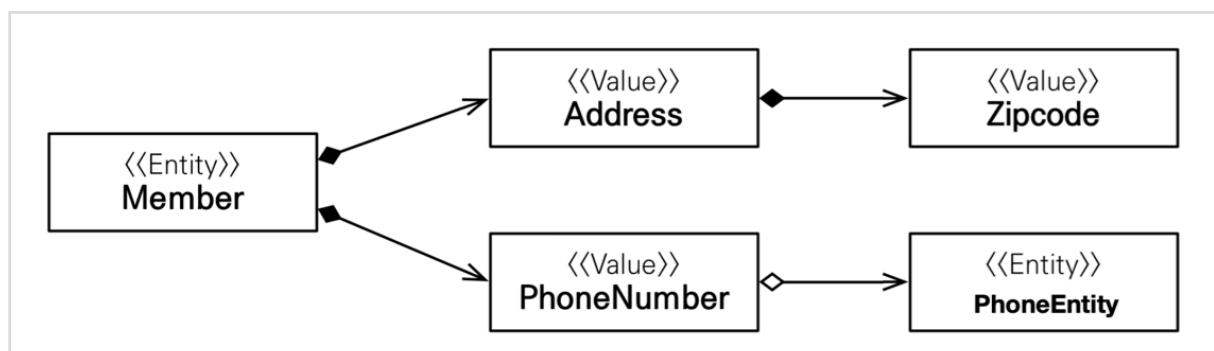
임베디드 타입을 사용해도 매핑하고 있는 테이블은 항상 같다!!

근데 왜 쓰냐?

객체와 테이블을 아주 세밀하게 매핑할 수 있음.

잘 설계한 ORM 어플리케이션은 매핑한 테이블보다 매핑한 클래스 수가 더 많음.

### 주의



임베디드 타입인 `PhoneNumber`가 엔티티인 `PhoneEntity`를 가짐.

`PhoneNumber`가 `PhoneEntity`의 fk만 가지면

## 한 엔티티에서 같은 값을 많이 사용하기

@AttributeOverrieds(@AttributeOverride(속성, 컬럼))로 재정의 가능.

#JPA