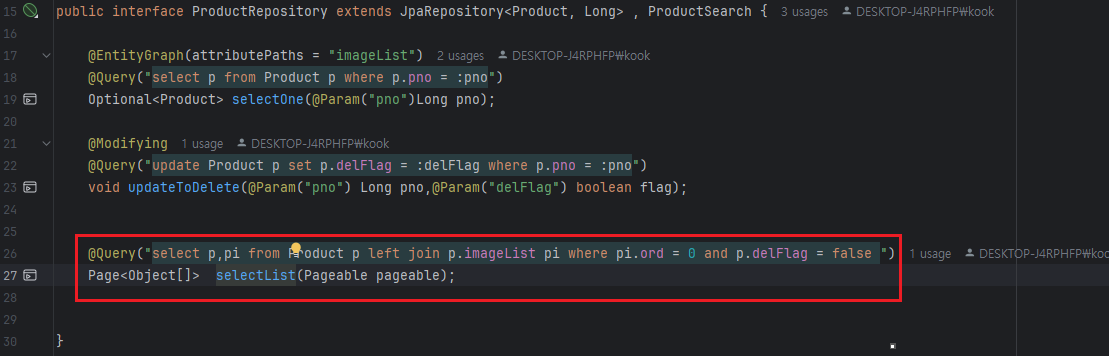
**#상품 목록 처리(서비스 계층)**

****

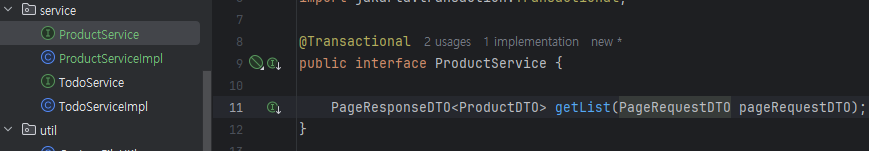
**서비스 계층에서 상품 목록을 가져올때 사용할**

**selectList의 결과 값.**

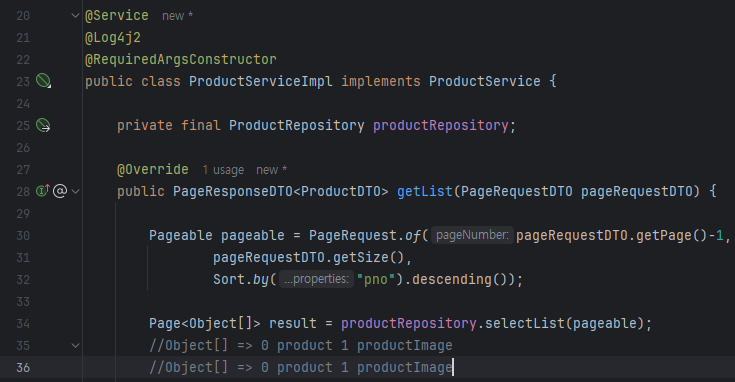
**(상품과 상품의 대표이미지 하나)들로 구성  
서비스 계층에서 이것들을 ProductDTO로 구성해줘야한다.**

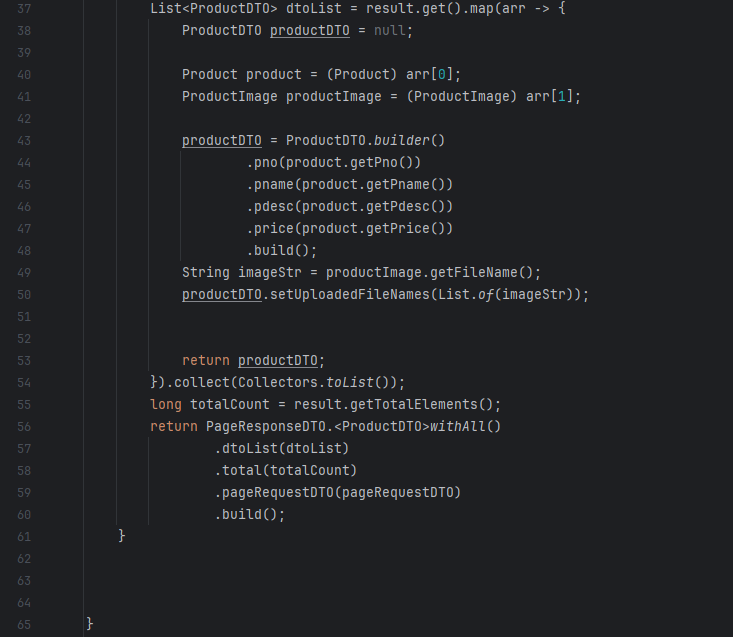
1. **ProductService 인터페이스를 작성**

**목록을 가져올 때 사용할 getList 정의**

****

1. **ProductServiceImpl 서비스 계층 코드 작성(getList)**

****

****

**페이징 처리를 위한 pageable 객체 정의하고 selectList 호출**

**호출 결과는 객체 배열로 0번째들에는 product 엔티티, 1번째들은 productImageList의 0번째를 가짐. (이미지 리스트의 0번째)**

**그것들을 DTO로 매핑 해준다.**

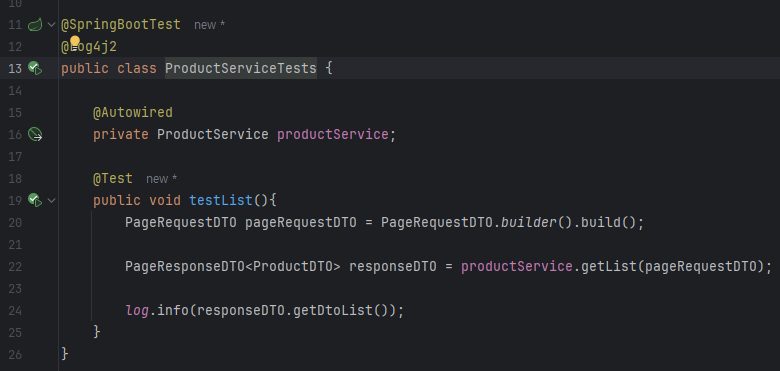
**그 과정에서 List.of(imageStr)을 해준 이유는 selectList로 가져오는 상품들 하나당 이미지도 하나씩을 가져오기 때문(하나의 상품당 다수의 이미지를 리스트로 가지지만)**

**하나의 문자열을 List으로 구성하여 넘겨주는 셈이다.**

**그런다음 PageResponseDTO에 @Builder어노테이션을 사용해 withAll이라고 명명해놓은 빌더용 메서드를 사용하여 PageResponseDTO를 구성하여 리턴해준다.**

**상품들의 리스트와 각 상품들의 0번째 이미지들을 구성하여 리턴해주는 것이다.**

**테스트 코드**

****

**PageRequestDTO의 멤버변수 page, size는 default값을 1, 10으로 각각 설정해둔 상태이기 때문에 default로 빌드하여 getList를 호출하였다.**

**responseDTO.getDtoList()를 로그로 찍어보면 아래와 같다.**

