

● Paging 처리

: 페이징을 위한 기본 작업이 정렬이다.

: 상속받은 JpaRepository를 따라가면 findAll 에서도 Pageable을 받아 처리함을 알 수 있다.

```
@Test
@DisplayName("Paging Test")
void pagingTest(){
    System.out.println("페이지 = 0, 페이지당 리스트 수 : 5");
    usersRepository.findAll(
        PageRequest.of(0, 5, Sort.by(Sort.Order.desc("name"))))
        .getContent().forEach(users -> System.out.println(users));

    System.out.println("페이지 = 1, 페이지당 리스트 수 : 5");
    usersRepository.findAll(
        PageRequest.of(1, 5, Sort.by(Sort.Order.desc("name"))))
        .getContent().forEach(users -> System.out.println(users));

    System.out.println("페이지 = 2, 페이지당 리스트 수 : 5");
    usersRepository.findAll(
        PageRequest.of(2, 5, Sort.by(Sort.Order.desc("name"))))
        .getContent().forEach(users -> System.out.println(users));

}
```

- Repository

```
//주어진 아이디보다 큰 데이터를 내림차순으로 검색 후 페이징 처리
Page<Users> findByIdGreaterThanOrderByDesc(Long id, Pageable paging);
```

- Test code

```
@Test
@DisplayName("findByIdGreaterThanOrderByDesc Paging 테스트")
void findByIdGreaterThanOrderByDescTest(){
    Pageable pageable = PageRequest.of(0, 5);
```

```

Page<Users> result = usersRepository
    .findByldGreaterThanOrderByldDesc(0L, pageable);
result.getContent().forEach(user -> System.out.println(user));
//총 페이지 수
System.out.println("Total Pages : " + result.getTotalPages());

//전체 데이터(엔티티) 개수
System.out.println("Total Counts : " + result.getTotalElements());

//현재 페이지 번호(0부터 시작)
System.out.println("Page Number : " + result.getNumber());

//페이지 당 데이터 개수
System.out.println("Page Size : " + result.getSize());

//다음 페이지 존재 여부
System.out.println("Has Next Page? : " + result.hasNext());

//시작 페이지(0) 존재 여부
System.out.println("Is First Page? : " + result.isFirst());
}

```

- Repository

```

//주어진 Color 값을 갖는 데이터를 추출
Page<Users> findByLikeColor(String color, Pageable pageable);

```

- Test code

```

@Test
@DisplayName("findByLikeColor 정렬 추가 Paging 테스트")
void findByLikeColorAndSortingTest(){
    Sort sort = Sort.by("name").descending();
    Pageable pageable = PageRequest.of(0, 5, sort);
    Page<Users> result = usersRepository.findByLikeColor("Orange", pageable);
    result.getContent().forEach(user -> System.out.println(user));
}

```

```
//총 페이지 수
System.out.println("Total Pages : " + result.getTotalPages());

//전체 데이터(엔티티) 개수
System.out.println("Total Counts : " + result.getTotalElements());

//현재 페이지 번호(0부터 시작)
System.out.println("Page Number : " + result.getNumber());

//페이지 당 데이터 개수
System.out.println("Page Size : " + result.getSize());

//다음 페이지 존재 여부
System.out.println("Has Next Page? : " + result.hasNext());

//시작 페이지(0) 존재 여부
System.out.println("Is First Page? : " + result.isFirst());
}
```