

From Miricanvas



**유효성 검사는
언제 하나요?**

PLAY!

QUESTION

Q

**나중에 쓰려고 저장하는 매개변수!
언제 유효성 검사를 할까요?**

나중에 사용할 때 검사한다.

지금 당장 검사한다.

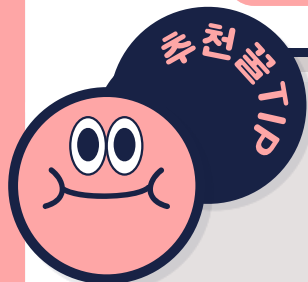
1: 나중에!



지금 할 일을 나중에 미루고 있지는 않나요?



나중에 검사해도 충분해! 문제 없어! 긍정적인 성격



예외가 발생하는 상황을 생각해봅시다.

List 형태의 파라미터를 입력받았다고 가정해봅시다.
null 체크를 안해서 필드에 null이 저장되어 버렸네요!
나중에 List를 사용하는 메소드에서 List.get(1)을 하려고 하니
NullPointerException이 발생했습니다.

이미 그 사이에 실행된 로직이 엄청 많은데...
언제 다 살펴보죠? π.π



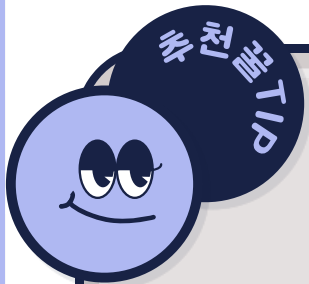
2. 지금 당장!



당신은 할 일을 미루지 않고 바로바로 하는 타입!



즉각적이고 깔끔한 방식을 선호



유효성 검사에도 예외는 있어요!

유효성 검사 비용이 지나치게 높거나,
실용적이지 않거나,
계산 과정에서 암묵적으로 검사가 수행되는 경우도 있어요!

`Collections.sort(list)`에서 `list`의 객체들은
모두 상호 비교될 수 있는 타입이어야 하죠.
하지만 우리는 이에 대해 검사를 하지는 않습니다.



REPLAY



**파라미터의 null 체크,
어떻게 하시나요?**

@Nullable 같은 어노테이션 사용하기

Objects.requireNonNull() 사용하기

1: @Nullable



어노테이션 쓰는걸 좋아해요!



특정 매개변수는 null이 될 수 있다고 알려줘요.



추천 TIP

**null이 사용될 수 있다고 암시는 해주지만
방지하기 까지는 어려울 수 있다.**

@Nullable은 null이 사용 가능한거지
사용 방지를 하지는 못합니다.

특정 라이브러리를 필요로 하기도 하고
표준적인 방법으로 쓰이고 있지는 않아요



REPLAY

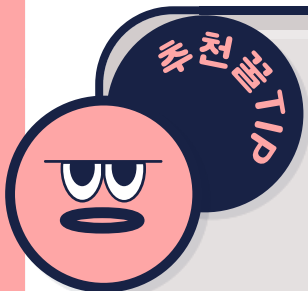
2. .requireNonNull()



null 검사를 수동으로 하기 싫은 당신에게 딱!



잘 만들어진 메소드를 사용하는 당신 엄지척 b



Object.requireNonNull()에 대해

Java 7에서 추가된 메소드로 유연하고, 사용하기 편해요!
더 이상 null 검사를 수동으로 하지 않아도 되니 ㄱㅇㄷ~~~

Java 9에서는 Objects에 범위 검사 기능도 더해졌어요

checkFromIndexSize()

checkFromToIndex()

checkIndex()



REPLAY

TIP!!!

단언문으로 검사하기



단언문 = assert



자신이 단언한 조건이 무조건 참이라고 선언



AssertionError!!!

런타임에 아무런 효과도, 성능 저하도 없어요

```
private static void sort(long a[], int offset, int length) {  
    assert a != null;  
    assert offset >= 0 && offset <= a.length;  
    assert length >= 0 && length <= a.length - offset;  
    // 계산 수행  
}
```