

# [아이템 49] 매개변수가 유효한지 검사하라

≡ 태그	6주차
장	8장 : 메서드

매개변수 제약은 반드시 문서화해야 하며 메서드가 시행되기 전 검사를 해야 합니다.

왜? 잘못된 값이 넘어오면 예상치 못한 오류가 발생하니까, 깔끔하게 예외를 던져주는 게 좋습니다.

자바 개발의 핵심적인 기본 원칙은 바로 명료성과 단순성입니다.

1. 컴포넌트는 정해진 동작이나 예측할 수 있는 동작만 수행해야 한다.
2. 컴포넌트는 가능한 작되, 그렇다고 너무 작아서 안된다.
3. 코드는 복사되는 게 아니라 재사용되어야 한다.
4. 컴포넌트 사이의 의존성은 최소로 유지해야 한다.
5. 오류는 만들어지자마자 가능한 한 빨리 잡아야 한다. (= 되도록 컴파일타임에 해결되어야 한다.)

이 규칙에 따르면 사실상 최고의 모범 사례를 만드는 것에 가깝겠죠. 그를 위하여 매개변수의 유효성 검사에 대해 충분히 검토가 이루어져야 합니다.

## javadoc 명세화

javadoc과 같은 정확한 명세가 존재할 때, 효율성과 시간 단축을 도모할 수 있습니다. 개발자들 사이의 올바른 작동을 보장하는데 중요한 역할을 합니다.

메서드의 시작 부분에서 매개변수의 유효성을 검사하고, 유효하지 않은 경우 적절한 예외를 던집니다.

```
/**
 * 매개변수의 유효성을 검사하고 처리하는 예제 메서드입니다.
```

```

*
* @param parameter 검사할 매개변수. 0에서 100 사이어야 합니다.
* @throws IllegalArgumentException 매개변수가 유효하지 않을 경우 발생합니다.
*/
public void someMethod(int parameter) {
    if (parameter < 0 || parameter > 100) {
        throw new IllegalArgumentException("매개변수는 0에서 100 사이어야 합니다.");
    }
    // 매개변수가 유효한 경우 메서드의 나머지 부분을 처리합니다.
}

```

가능한 전제조건을 `@param` 에 넣고, 사후 조건을 `@return` 절 혹은 `@throws` 절에 넣어주며 문서화 주석을 추가합니다.

이러한 예외 문서화는 public이거나 protected인 경우 더욱 중요!

## assert문

`assert` 문을 사용하여 매개변수 유효성을 검사하는 경우는 보통 디버깅 목적으로 사용된다 고👁️.

개발자가 코드의 특정 지점에서 조건을 검증하고, 조건이 거짓인 경우 `AssertionError`를 발생시키도록 하는데 사용됩니다.

```

public void someMethod(int parameter) {
    assert parameter >= 0 && parameter <= 100 : "매개변수는 0에서 100 사이어야 합니다.";

    // 매개변수가 유효한 경우 메서드의 나머지 부분을 처리합니다.
}

```

이 경우, `assert` 문은 개발자가 프로그램을 디버깅할 때만 유효하며, 런타임에는 비활성화될 수 있습니다. 따라서 실제로 사용자에게 제공되는 라이브 코드에서는 `assert` 문을 사용하여 매개변수 유효성을 검사하는 것은 권장되지 않습니다.

## 매개변수에 제약을 두는게 좋은건가?

잘못된 동작을 방지한다는 측면에서 안전성이 보장되고, 사용자에게 유효값을 정확히 알려 주니 가독성도 좋고.. 제약 두는게 오히려 좋은거 아닌가? 🤔

⇒ X

사용자에게 좋은 경험을 제공한다는 생각에 매몰되어 지나치게 엄격하게 설정하게 되면 악 효과가 발생할 수 있습니다.

합리적인 범위 내에 제약을 두고, 반드시 필요한 경우에만 예외를 던지게끔 처리하는 것이 사용자가 쉽게 이해하고 사용할 수 있는 프로그래밍 방법이라 생각합니다.