이펙티브자바 ch7.

람다와 스트림

Item 43.

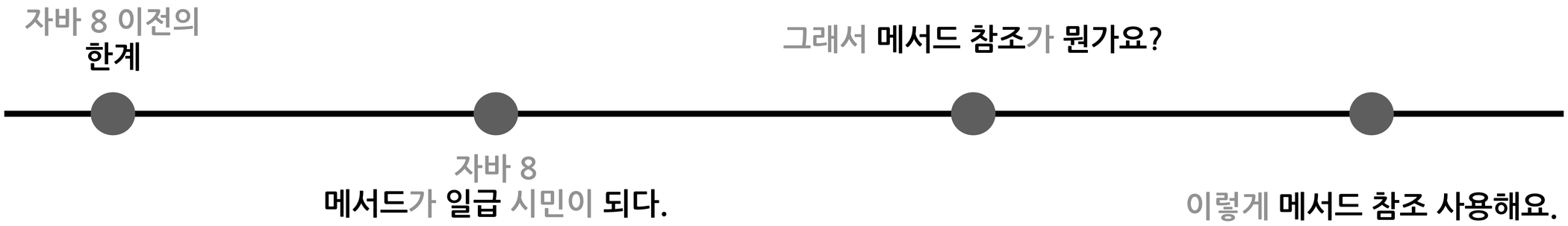
람다보다는 메서드 참조를 사용하라

2025.06.09

김진수

람다보다는 메서드 참조를 사용하라

목차.



람다보다는 메서드 참조를 사용하라

자바 8 이전의 한계

‘일급 시민’과 ‘이급 시민’ 이란?

일급 시민(First-class citizen)

프로그래밍 언어에서 **값**처럼 자유롭게 다룰 수 있는 요소

역사적으로 그리고 전통적으로 프로그래밍 언어에서는 자유롭게 다룰 수 있는 값을 일급 (first-class 퍼스트 클래스)값이라고 부른다.(first-class 일급은 1960년대 미국 시민 권리에서 유래)

람다보다는 메서드 참조를 사용하라

자바 8 이전의 한계

‘일급 시민’과 ‘이급 시민’ 이란?

일급 시민(First-class citizen)

프로그래밍 언어에서 **값**처럼 자유롭게 다룰 수 있는 요소

int, double 같은 **기본 타입**

new String(“dog”) **객체 참조 값**

역사적으로 그리고 전통적으로 프로그래밍 언어에서는 자유롭게 다룰 수 있는 값을 일급 (first-class 퍼스트 클래스)값이라고 부른다.(first-class 일급은 1960년대 미국 시민 권리에서 유래)

람다보다는 메서드 참조를 사용하라

자바 8 이전의 한계

‘일급 시민’과 ‘이급 시민’ 이란?

일급 시민(First-class citizen)

프로그래밍 언어에서 **값**처럼 자유롭게 **다룰 수 있는** 요소

int, double 같은 **기본 타입**

new String(“dog”) **객체 참조 값**

변수에 담기

메서드 인자로 전달

반환 값으로 사용

역사적으로 그리고 전통적으로 프로그래밍 언어에서는 자유롭게 다룰 수 있는 값을 일급 (first-class 퍼스트 클래스)값이라고 부른다.(first-class 일급은 1960년대 미국 시민 권리에서 유래)

람다보다는 메서드 참조를 사용하라

자바 8 이전의 한계

‘일급 시민’과 ‘이급 시민’ 이란?

이급 시민(Second-class citizen)

값처럼 자유롭게 전달할 수 없는 요소

메서드나 클래스

람다보다는 메서드 참조를 사용하라

자바 8 이전의 한계

메서드를 직접 전달할 수 없던 시절

Ex) List의 문자열 정렬

```java
public static void main(String[] args) {
    List<String> words = Arrays.asList("Apple", "banana", "Cherry");

    // Java 8 이전 방식
    Collections.sort(words, new Comparator<String>() {
        @Override
        public int compare(String s1, String s2) {
            return s1.compareToIgnoreCase(s2);
        }
    });
}
```

람다보다는 메서드 참조를 사용하라

자바 8 이전의 한계

메서드를 직접 전달할 수 없던 시절

Ex) List의 문자열 정렬

```java
public static void main(String[] args) {
    List<String> words = Arrays.asList("Apple", "banana", "Cherry");

    // Java 8 이전 방식
    Collections.sort(words, new Comparator<String>() {
        @Override
        public int compare(String s1, String s2) {
            return s1.compareToIgnoreCase(s2);
        }
    });
}
```

```
new String().com
    m compareTo(String anotherString)
    m compareToIgnoreCase(String str)
    Press ⏎ to insert, ⇥ to replace  Next Tip
```

람다보다는 메서드 참조를 사용하라

자바 8. 메서드가 일급 시민이 되다.

메서드 참조(::)의 등장

```java
public static void main(String[] args) {
    List<String> words = Arrays.asList("Apple", "banana", "Cherry");

    // Java 8 메서드 참조 방식
    words.sort(String::compareToIgnoreCase);
}
```

람다보다는 메서드 참조를 사용하라

그래서 메서드 참조가 뭔가요?

메서드 참조 정의

그래서 메서드 참조가 뭔가요?

람다보다는 메서드 참조를 사용하라

그래서 메서드 참조가 뭔가요?

```
람다 표현식: (Apple a) -> a.getWight()
```

"Apple 객체 a가 주어지면, a의 getWeight() 메서드를 호출해라."
(어떻게)

람다보다는 메서드 참조를 사용하라

그래서 메서드 참조가 뭔가요?

```
람다 표현식: (Apple a) -> a.getWight()
```

"Apple 객체 a가 주어지면, a의 getWeight() 메서드를 호출해라.”
(어떻게)

```
메서드참조:   Apple::getWeight
```

"Apple의 getWeight 메서드를 사용해라.”
(무엇을)

람다보다는 메서드 참조를 사용하라

목차.

이렇게 메서드 참조 사용해요.

📌 유형 1. 정적 메서드 참조

📌 유형 2. 특정 타입의 인스턴스 메서드 참조

📌 유형 3. 기존 객체의 인스턴스 메서드 참조

람다보다는 메서드 참조를 사용하라

이렇게 메서드 참조 사용해요.

---

1. 정적 메서드 참조 (Static Method Reference)

   클래스명 :: 정적메서드명

람다 (String s) -> Integer.parseInt(s)

메서드 참조 Integer::parseInt

```
// 람다
                                .map(s -> Integer.parseInt(s))

// 메서드 참조
                                .map(Integer::parseInt)
```

람다보다는 메서드 참조를 사용하라

이렇게 메서드 참조 사용해요.

2. 특정 타입의 인스턴스 메서드 참조

타입명 :: 인스턴스메서드명

람다 (String s) -> s.length()

메서드 참조 String::length

```
// 람다
                                    .map(s -> s.length())

// 메서드 참조
                                    .map(String::length)
```

람다보다는 메서드 참조를 사용하라

이렇게 메서드 참조 사용해요.

3. 기존 객체의 인스턴스 메서드 참조

객체참조변수명 :: 인스턴스 메서드명 List<String> list = Arrays.asList(“apple”…);

람다 Item -> list.contains(item);

메서드 참조 list::contains;

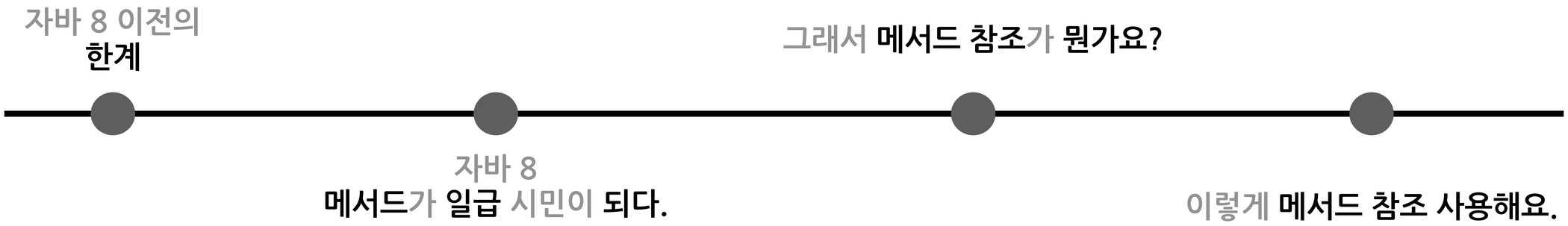
```
List<String> list = Arrays.asList("apple", "banana", "cherry");

// 람다식 방식
Predicate<String> lambda2 = item -> list.contains(item);

// 메서드 참조 방식
Predicate<String> methodRef2 = list::contains;
```

람다보다는 메서드 참조를 사용하라

목차.



이펙티브자바 ch7.

람다와 스트림

Item 43.

람다보다는 메서드 참조를 사용하라

2025.06.09

김진수