

# 아이템 26. 로 타입은 사용하지 말라

🕒 생성일	@2021년 10월 31일 오후 7:45
☰ 태그	

- 제네릭 클래스, 제네릭 인터페이스
  - 클래스와 인터페이스 선언에 타입 매개변수가 쓰일 경우 부르는 명칭
- 로(raw) 타입이란?
  - 제네릭 타입에서 타입 매개변수를 전혀 사용하지 않을 때를 말한다
  - ex) List<E> 의 로 타입은 List 이다. (타입 매개변수 전혀 사용안함)

ex) 로 타입으로 컬렉션을 사용할 경우

```
private final Collection stamps = ...;
```

```
stamps.add(new Coin(...)); //도장을 넣을 컬렉션인데 코인을 넣었다고 한다면..?
```

```
// 컬렉션에서 이 코인을 다시 꺼내기 전 까지는 알 수가 없다. 아주 안좋은 방식
```

- 로(raw) 타입을 쓰면 제네릭이 안겨주는 안전성과 표현력을 모두 잃게 되므로 절대 써서는 안된다.
  - List 같은 로타입은 사용해서는 안되나, List<Object>같은 임의 객체를 사용하는건 괜찮다?
    - 로 타입의 경우 제네릭 타입에서 완전 벗어난 경우, 하지만 List<Object> 의 경우 컴파일러에게 모든 타입을 허용하겠다고 의사를 전달한 것
- 비한정적 와일드카드 타입과 로 타입의 차이는 무엇인가?
  - 비한정적 와일드카드 <?> :어떤 타입이라도 담을 수 있는 가장 범용적인 매개변수화
  - 와일드카드 타입은 안전하고, 로 타입은 안전하지 않다.
    - (계속 나오는 얘기인데, 로 타입은 컴파일 시점에서 문제를 알 수없다.)

- 로타입을 쓰지말라는 규칙에도 예외는 존재 한다.
  - class 리터럴에는 로 타입을 써야 한다.
    - 자바 명세는 class 리터럴에 매개변수화 타입을 사용하지 못하게 했다.( 배열과 기본 타입은 허용한다.)
      - List.class, String[].class 는 허용, List<String>.class , List<?>.class 는 허용하지 않음
  - instanceof 연산자
    - 런타임에는 제네릭 타입 정보가 지워지므로 instanceof 연산자는 비한정적 와일드카드 타입 외 매개변수화 타입에는 적용 할 수 없다.
    - 비한정적 와일드카드와 로 타입은 똑같이 동작하므로 <?>식의 지저분한 코드 대신 로 타입을 쓰는 편이 깔끔하다.
      - ex) if( o instanceof Set) {...}
- 정리
  - Set<Object> 는 어떤 타입의 객체도 저장할 수 있는 매개변수화 타입
  - Set<?> 는 모종의 타입 객체만 저장할 수 있는 와일드카드 타입
  - 로 타입은 제네릭 타입 시스템에 속하지 않고 안전하지 않다.

## • (추가)

### 제네릭

: 지금은 이 타입을 모르지만, 이 타입이 정해지면 그 타입 특성에 맞게 사용하겠다!

### 와일드 카드

: 지금도 이 타입을 모르고, 앞으로도 모를 것이다!

출처:

<https://vvshinevv.tistory.com/55>