

# Effective Java

## 아이템 28

'배열'보다는 **리스트**를 사용하라'

## '배열'보다는 **리스트**를 사용하라'

### ▸ 배열과 리스트

1. 타입을 구분하는 기준
2. 제네릭 시스템의 목적 & 타입 인지 시점
3. 실체화 불가 타입 & 실체화 타입

### 참고

- The Java Tutorials - Generics
- The Java Tutorials - Type Erasure
- Type Erasure in Java Explained
- Java Language Specification: Type Erasure
- Why following types are reifiable & non-reifiable in java?

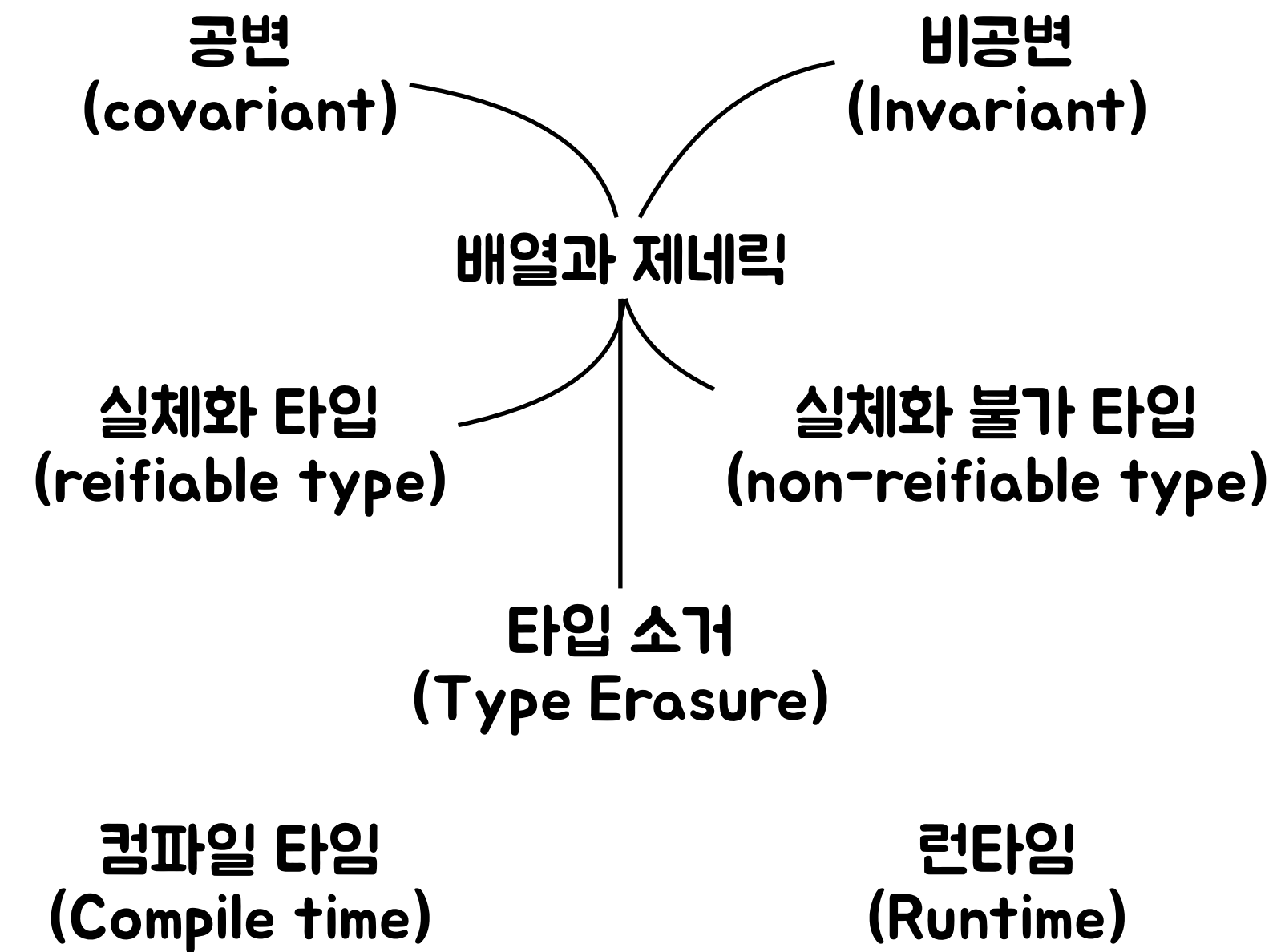
## '배열보다는 리스트를 사용하라'

```
Object obj = new String();
```

```
Object[] objArray = new String[len];
```

```
List<Object> list = new ArrayList<Long>();
```

```
List list = new ArrayList();
```



### 배열과 리스트

1. 타입을 구분하는 기준
2. 제네릭 시스템의 목적 & 타입 인지 시점
3. 실체화 불가 타입 & 실체화 타입

배열과 리스트가 '타입을 구분하는 기준'

- 배열과 리스트
- 1. 타입을 구분하는 기준
- 2. 제네릭 시스템의 목적 & 타입 인지 시점
- 3. 실체화 불가 타입 & 실체화 타입

배열: 'Object 타입의 여러 변수를 하나의 묶음으로 다루는 것'

공변

```
Object object = new Long();  
Object[] objects = new Long[3];
```

실체화 타입 (런타임 오류)

```
objects[0] = '아이템';
```

컬렉션 프레임워크: '데이터 그룹을 다루고 표현하기 위한 단일화된 구조'

제네릭: 다양한 타입의 객체들을 다루는 메서드나 컬렉션 클래스에 컴파일 시의 타입 체크를 해주는 기능

불공변 (컴파일 타임 오류)

```
List<String> strings = new ArrayList<>();  
List<String> strings = new ArrayList<Long>();
```

실체화 불가 타입

```
strings.add('문자열');
```

7버전 부터 추정이 가능한 경우 타입을 생략할 수 있다.

배열과 리스트가 '타입을 구분하는 기준'

한글 용어	영문 용어	예	아이템
매개변수화 타입	parameterized type	List<String>	아이템 26
실제 타입 매개변수	actual type parameter	String	아이템 26
제네릭 타입	generic type	List<E>	아이템 26, 29
정규 타입 매개변수	formal type parameter	E	아이템 26
비한정적 와일드카드 타입	unbounded wildcard typ	List<?>	아이템 26
로 타입	raw type	List	아이템 26
한정적 타입 매개변수	bounded type parameter	<E extends Number>	아이템 29
재귀적 타입 한정	recursive type bound	<T extends Comparable<T>>	아이템 30
한정적 와일드카드 타입	bounded wildcard type	List<? extends Number>	아이템 31
제네릭 메서드	generic method	static <E> List<E> asList(E[] a)	아이템 30
타입 토큰	type token	String.class	아이템 33

배열과 리스트

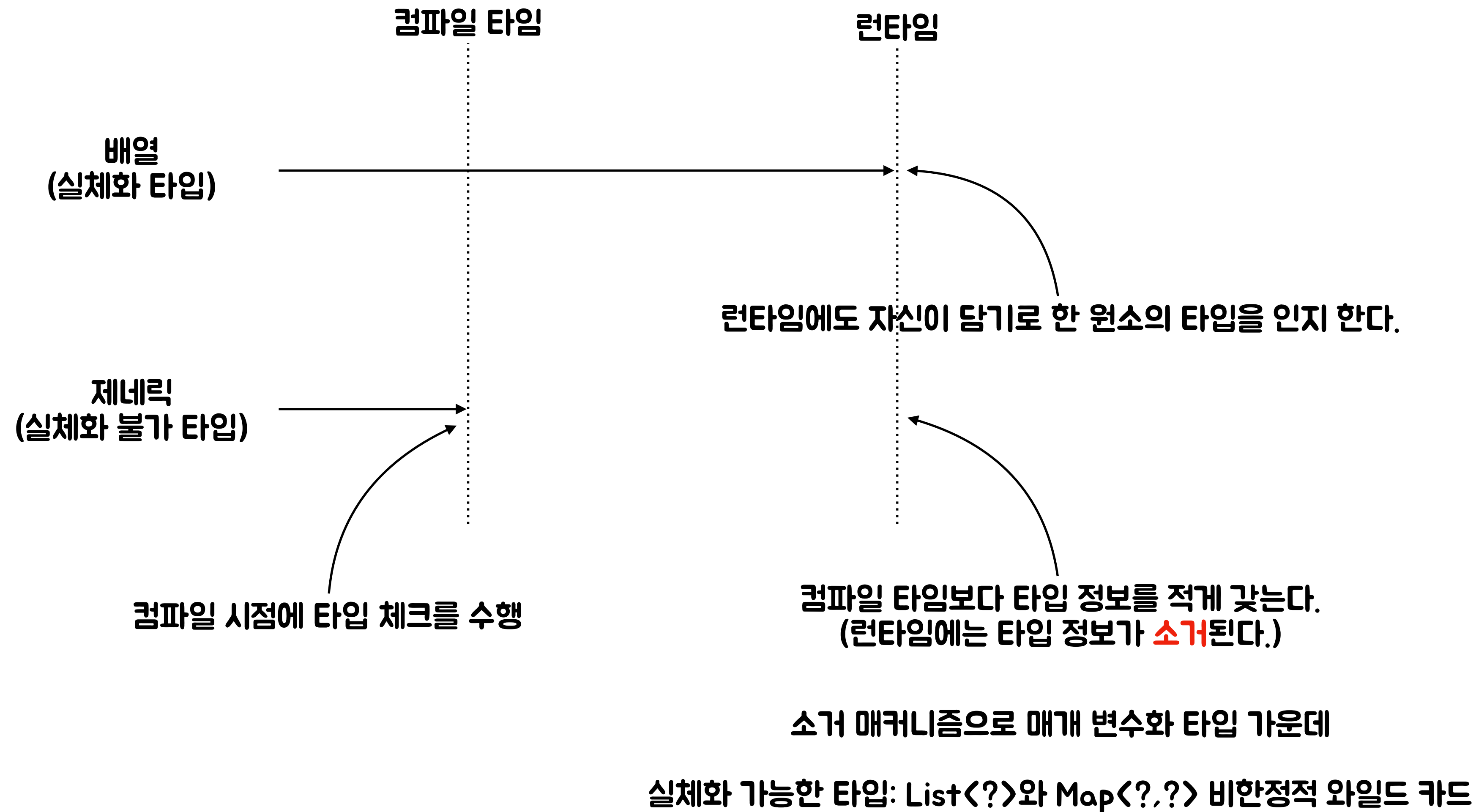
- 1. 타입을 구분하는 기준
- 2. 제네릭 시스템의 목적 & 타입 인지 시점
- 3. 실체화 불가 타입 & 실체화 타입

<div data-bbox="1026 45 1559 101">제네릭 타입 시스템의 목적</div>	<div data-bbox="2618 45 2918 101">▸ 배열과 리스트</div> <div data-bbox="2618 146 3292 375"><div data-bbox="2618 146 3008 193">1. 타입을 구분하는 기준</div><div data-bbox="2618 238 3292 285">2. 제네릭 시스템의 목적 &amp; 타입 인지 시점</div><div data-bbox="2618 330 3185 375">3. 실체화 불가 타입 &amp; 실체화 타입</div></div>
<div data-bbox="629 911 1952 968">'런타임에 <code>ClassCastException</code>이 발생하는 일을 막는 역할을 한다'</div>	

## 배열과 제네릭의 타입 인지 시점

### 배열과 리스트

1. 타입을 구분하는 기준
2. 제네릭 시스템의 목적 & 타입 인지 시점
3. 실체화 불가 타입 & 실체화 타입





식별화 타입과 식별화 불가 타입		배열과 리스트
식별화 타입 (reifiable type)	A primitive type (such as int)	1. 타입을 구분하는 기준
	A nonparameterized class or interface type (such as Number, String, or Runnable)	2. 제네릭 시스템의 목적 & 타입 인지 시점
	A parameterized type in which all type arguments are unbounded wildcards (such as List<?>, ArrayList<?>, or Map<?, ?>)	3. 실체화 불가 타입 & 실체화 타입
	A raw type (such as List, ArrayList, or Map)	
	An array whose component type is reifiable (such as int[], Number[], List<?>[], List[], or int[][])	
식별화 불가 타입 (non-reifiable type)	A type variable (such as T)	
	A parameterized type with actual parameters (such as List<Number>, ArrayList<String>, or Map<String, Integer>)	
	A parameterized type with a bound (such as List<? extends Number> or Comparable<? super String>)	