

Java Generic 이란?

우선 Generic의 사전적인 의미는 "포괄적인, 총칭의, 회사 이름이 붙지 않은, 일반 명칭으로 판매되는" 즉, 딱 하나를 정하지 않고 범용적이고 포괄적이라는 의미입니다.

Java 에서 Generic은?

Data type을 특정한 type 하나로 정하지 않고 사용할 때마다 변경 할 수 있게 범용적이고 포괄적으로 지정한다는 의미입니다.

그렇다면 왜 사용해야 할까?

제일 큰 이유는 Object 클래스의 한계 때문입니다.

```
List list = new ArrayList();  
list.add("hello");  
String str = list.get(0);  
System.out.println(list);
```

위의 코드는 정상으로 작동 할까요?

아닙니다, 3번째 줄에서 오류가 발생합니다. list의 데이터 타입은 Object인데, 3번째 줄에서는 데이터타입이 String이기 때문입니다.

그래서 제대로 동작하게 하려면,

```
String str = (String)list.get(0);
```

형변환 즉, Casting을 해야합니다.

"아니 그러면 그냥 Casting을 하면 되네?"

반은 맞고 반은 틀립니다. 왜냐하면 시스템 성능의 문제가 생기기 때문입니다. 지금은 이렇게 Casting할 수 있지만, 만약 데이터가 수만개가 넘어간다면 모든 데이터를 하나하나 casting해야하기 때문입니다.

"그러면 어떻게 해야 되나요?"

그래서 사용하는 것이 Generic입니다. 위의 코드를 인용한다면,

```
List<String> list = new ArrayList<>();  
list.add("hello");  
String str = list.get(0);
```

```
System.out.println(list);
```

이렇게 코드를 작성한다면 별도의 Casting없이 동작합니다.

Generic의 장점

- 1.제네릭을 사용하면 잘못된 타입이 들어올 수 있는 것을 컴파일 단계에서 방지 가능하다
- 2.클래스 외부에서 타입을 지정해주기 때문에 따로 타입을 체크하고 캐스팅 해줄 필요가 없다.
- 3.비슷한 기능을 지원한느 경우 코드의 재사용성이 높아진다.

Generic의 단점

- 1.제네릭은 클래스와 인터페이스만 적용되기 때문에 기본타입(Primitive Type)은 사용할 수 없다.
- 2.제네릭을 타입을 사용하여 객체르 생성 불가
- 3.배열에 대한 제한(제네릭 클래스 또는 인터페이스 타입의 배열을 선언할 수 없다)
(단, 제네릭 타입의 배열 선언은 가능)

ex) `Class<Integer>[] arr1 = new Class<>[10]; // error`
`Class<Integer>[] arr2 = new Class[10]; // Okay`

여담으로, Generic은 계속 형 안정성을 제공하는 것이 아닙니다.
컴파일 시점에만 제공하고, 런타임 시점에는 형 안정성을 제공하지 못합니다. 유의하고 사용하세요.

컴파일타임 : 개발자에 의해 개발 언어로 소스코드가 작성되며, 컴파일 과정을 통해 컴퓨터가 인식할 수 있는 기계어 코드로 변환되어 실행 가능한 프로그램이 되는 과정

런타임 : 컴파일 과정을 마친 응용 프로그램이 사용자에게 의해 실행되어 지는 때 (time)

참고 블로그

[https://medium.com/](https://medium.com/%EC%8A%AC%EA%B8%B0%EB%A1%9C%EC%9A%B4-%EA%B0%9C%EB%B0%9C%EC%83%9D%ED%99%9C/java-generic-%EC%9E%90%EB%B0%94-%EC%A0%9C%EB%84%A4%EB%A6%AD-f4343fa222df)

[%EC%8A%AC%EA%B8%B0%EB%A1%9C%EC%9A%B4-%EA%B0%9C%EB%B0%9C%EC%83%9D%ED%99%9C/java-generic-%EC%9E%90%EB%B0%94-%EC%A0%9C%EB%84%A4%EB%A6%AD-f4343fa222df](https://medium.com/%EC%8A%AC%EA%B8%B0%EB%A1%9C%EC%9A%B4-%EA%B0%9C%EB%B0%9C%EC%83%9D%ED%99%9C/java-generic-%EC%9E%90%EB%B0%94-%EC%A0%9C%EB%84%A4%EB%A6%AD-f4343fa222df)

<https://st-lab.tistory.com/153>

<https://madplay.github.io/post/java-generic-advanced>

[https://m.blog.naver.com/PostView.naver?
isHttpsRedirect=true&blogId=gnggh0101&logNo=221254536442](https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=gnggh0101&logNo=221254536442)

<https://dd-corp.tistory.com/9>