제네릭 (GENERIC)

제네릭이란

- ‘일반적인’ 이라는 뜻처럼 상황에 따라 다양한 데이터 타입으로 사용할 수 있는 형식

- 예를 들어 어떤 인스턴스를 담는 **box**라는 클래스가 있다고 했을 때,
- 상황에 따라 정수도 담고 싶고, 문자열도 담고 싶다면...

제네릭 사용법

- 클래스를 선언할 때 이름 뒤에 <T>와 같이 선언
- ex) public class box<T> { ... }

- 이후 box 클래스의 인스턴스를 생성할 때 <T> 자리에 실제 타입을 대입해주면 됨
- ex) box<int> intBox = new box<>();

- 메서드에 사용할 때는 리턴 타입 앞에 <T>를 선언

주요 개념 (바운디드 타입)

- <T> 자리에 들어갈 수 있는 타입의 범위를 제한해주는 기능

- 1. extend - public class box<T extends Animal> { ... }
- 2. super - extend의 반대

주요 개념 (와일드 카드)

- **<?>** 와 같이 사용
- 주로 메서드의 매개변수에 사용
- 모든 타입이 가능하다는 의미

- 마찬가지로 **extend**와 **super**를 이용하여 범위 제한 가능

ERASURE (소거)

- 어떤 요소의 타입을 컴파일 과정에서만 검사하고 런타임에서는 알 수 없는 것

- 컴파일을 하게 되면 <T> 타입들이 모두 **Object** 형식으로 변하고, 이후 <T>에 실제로 값을 대입하는 선언부에서 **Object**를 실제 타입으로 형변환을 해줌

- 하한제한이 있으면 **Object**로, 상한 제한이 있으면 그 부모 클래스로 치환됨

QNA