

지네릭스

1. 지네릭스란?

2. 지네릭 클래스의 선언

- 컴파일 시 타입 변수 **<T>** 제거
- 객체 생성 시 **T → Apple** 형변환 발생
- 컴파일 시 **T → Object**
- 객체 생성하고 사용 시 **Object → Apple** 형변환
- **static** 멤버에 타입 변수 **T**를 사용 불가
 - T는 인스턴스 변수로 간주되기 때문이다.
- 지네릭 타입의 배열 생성 불가
- 타입 매개변수 → 제거 → **Object** : 컴파일 시
- 실 사용 시 : 인스턴스 생성(객체 생성)
형 변환 **Object → Apple**
- 타입 매개변수 → 제거 → **fruit** : 컴파일 시
- 형 변환 **Fruit → Apple**

3. 지네릭스의 용어

- 타입 변수
 - T(Type), E(Element), K(Key), V(Value)
- Class Box {}
 - **Box** : 지네릭 클래스, T의 Box 또는 T Box
 - **T** : 타입 변수 또는 타입 매개변수(T는 타입문자)
 - **Box** : 원시 타입
- Box b = new Box();
 - 지네릭타입 호출 : 타입 매개변수에 타입을 지정하는 것
 - 매개변수화된 타입 또는 대입된 타입 : String

- 컴파일 후에는 **Box**는 원시타입 **Box**로 바뀜.
→ 지네릭 타입 제거

4. 지네릭스의 제한

- **T extends Apple**
 - **extends Apple** 없으면 **T - Object**
 - **extends Apple** 있으면 **T - Apple**
- **extends** : 하한 제한, 하위 클래스 모두
- **super** : 상한 제한, 상위 클래스 모두

5. 지네릭 클래스의 객체 생성과 사용

- **static**

6. 제한된 지네릭 클래스

7. 와일드 카드 : ?

- 메서드 호출 시 에 자료형으로 형 변환
- **? → Object**
- **? extends T** : T 자료형 상한 제한 : T + 하위클래스만 가능
- **? super T** : T 자료형 하한 제한 : T + 상위클래스만 가능

8. 지네릭 메서드

```
public class Juicer<T> { // 인스턴스 만들 때 생성
    public static void make(Box<? extends Fruit> fruitBox) {
        fruitBox.printInfo();
        System.out.println("주스 완성");
    }
    // T는 호출될 때 생성
    // 위와 같이 복잡할 땐 아래와 같이 제네릭 메서드를 사용해 간편하게 사용
    public static <T extends Fruit> void make2(Box<T> fruitBox) {
        fruitBox.printInfo();
        System.out.println("주스 완성");
    }
}
```