

# Generics <> II

## 제네릭을 구성하는 다양한 문법적 요소 I

### 제네릭 메소드의 정의와 호출

- 클래스를 설계함에 있어서 일부 메소드만 제한적으로 제네릭화 할 수 있다.
- 메소드 정의 시 매개변수의 자료형을 결정하지 않는 문법
- 제네릭 메소드를 호출하는 순간 자료형을 결정해야 한다.

```
class InstanceTypeShower {
    int showCnt = 0;

    public <T> void showInstType(T inst){
        System.out.println(inst);
        showCnt;
    }

    void showPrintCnt(){ . . . . . }
}

class GenericsMethod {
    public static void main(String[] args){
        AAA aaa = new AAA();
        BBB bbb = new BBB();

        InstanceTypeShower shower = new InstanceTypeShower();

        shower.<AAA>showInstType(aaa);
        shower.<BBB>showInstType(bbb);

        shower.showPrintCnt();
    }
}
```



### 복수의 제네릭

- 하나의 타입정보 뿐만이 아니라 여러개의 타입정보를 복수의 제네릭 문자로 표현할 수 있다.
- T, U와 같은 제네릭 문자는 상징적이다. 따라서 다른 문자로 대체가능!

대체로 암묵적으로 정해져있는 약속이 있다.

- E - Element (자바의 컬렉션에서 널리 사용되고 있다.)
- K - Key
- N - Number
- T - Type
- V - Value
- S,U,V etc. - 2nd, 3rd, 4th types

```
class Person<T, S> {
    public T info;
    public S id;

    Person(T info, S id){
        this.info = info;
        this.id = id;
    }
}

public class GenericClass {
    public static void main(String[] args) {
        Person<EmployeeInfo, int> p1 = new Person<EmployeeInfo, int>(new EmployeeInfo(1), 1);
    }
}
```

---

```
class InstanceTypeShower2 {
    int showCnt = 0;

    public <T, U> void showInstType(T inst1, U inst2){
        System.out.println(inst1);
        System.out.println(inst2);
        showCnt;
    }

    void showPrintCnt(){ . . . . . }
}

class GenericsMethod {
    public static void main(String[] args){
        AAA aaa = new AAA();
        BBB bbb = new BBB();

        InstanceTypeShower2 shower = new InstanceTypeShower2();

        shower.<AAA, BBB>showInstType(aaa, bbb);
        shower.showPrintCnt();
    }
}
```

---

## 매개변수의 자료형 제한

- 제네릭으로 올 수 있는 데이터 타입을 특정 부모 클래스, 인터페이스의 자식으로 제한할 수 있다.
- class <T extends AAA> {

```
        . . . . .  
    }
```

- `public static <T extends AAA> void myMethod(T Param){ . . . . . }`

---

## 제네릭 메소드와 배열

- 다음과 같이 매개변수를 선언하면, 매개변수에 전달되는 참조값을 배열 인스턴스의 참조값으로 제한.

`T[] arr`

```
public static <T> void showArrayData(T[] arr){  
  
    for(int i=0; i<arr.length; i++){  
        System.out.println(arr[i]);  
    }  
  
}
```

---