

# 01.제네릭

[예 제]

C00Generic

C01Generic

## ■ 제네릭

### ■ 제네릭이란

- 클래스 내부에서 사용할 데이터 타입을 외부에서 지정하는 기법
- 미지정된 데이터타입이므로 여러 데이터타입의 자료형을 적용할수 있다

```
class Person <T>{  
    public T info;  
}  
Person<String> p1 = new Person<String>();  
Person <StringBuilder> p2 = new Person<StringBuilder>();
```

## 02.복수의 제너릭

### ■ 복수의 제너릭

- 여러개의 제네릭을 필요로 하는 경우 사용

<>연산자에 복수개의 제너릭 타입을 지정

```
class Person<T, S>{  
    public T info;  
    public S id;  
    Person(T info, S id){  
        this.info = info;  
        this.id = id;  
    }  
}
```

[예제]

C02Generic

전형적인 타입 매개변수

타입 매개변수	설명
E	원소 (Element)
K	키 (Key)
N	숫자 (Number)
T	타입 (Type)
V	값 (Value)

## 03.매서드 제너릭

### ■ 매서드 제너릭

- 리턴 타입 앞에 "<>" 기호를 추가하고 타입 파라미터를 기술한다

```
class Person<T, S>{  
    public T info;  
    public S id;  
    Person(T info, S id){  
        this.info = info;  
        this.id = id;  
    }  
    public <U> void printInfo(U info){  
        System.out.println(info);  
    }  
}
```

[예 제]

C03Generic

C04Pair

C04Util

C04CompareMethodEx

## 04.제너릭의 제한

### ■ extends

- 제네릭으로 올 수 있는 데이터 타입을 특정 클래스의 자식으로 제한할 수 있다.

```
abstract class Info{
    public abstract int getLevel();
}
class EmployeeInfo extends Info{
    public int rank;
    EmployeeInfo(int rank){ this.rank = rank; }
    public int getLevel(){
        return this.rank;
    }
}
class Person<T extends Info>{
    public T info;
    Person(T info){ this.info = info; }
}
```

Person의 T는 Info 클래스나 그 자식 외에는 올 수 없다.

[예 제]

C05Generic

C06GenericTest

C07GenericTest

C08GenericTest