

제 36강

컬렉션 프레임워크

교재:p222~224

목차

1. 컬렉션 프레임 워크

1. 컬렉션 프레임 워크?
2. 제네릭?

1. 컬렉션 프레임워크

컬렉션 프레임워크(Collection Framework)이란?

: 데이터를 효율적으로 다루기 위한 클래스들의 집합

- 자료구조(Data Structure)

: 자료들을 저장할 때, 효율적인 구조로 저장하는 것(데이터의 구조)

- 컬렉션프레임워크

: 자바에서 제공하는 자료구조

1. 컬렉션 프레임워크

컬렉션 프레임워크의 종류

<Collection 인터페이스를 구현하는 클래스>

- Set
- List
- Queue
- Map

iterator?

컬렉션 요소에 접근해서 반복적으로 요소 접근 시
도움을 주는 객체

<제공된 메서드>

| 메서드 | 설명 |
|----------------------|-------------------------|
| add(E e) | 지정요소(e) 추가 |
| contains(Object obj) | 객체 obj 존재 여부 |
| isEmpty() | 컬렉션 비어있는지 여부 |
| iterator() | 컬렉션의 iterator 반환 |
| remove(Object obj) | 객체 obj 제거 |
| size() | 요소 개수 반환 |

2. 제네릭(Generic)

제네릭(Generic)?

: 클래스가 다른 객체를 미리 명시하는 기법

의미? “제네릭 == 일반화”

ex) 택시를 다루던, 버스를 다루던 그냥 자동차(일반화) 라고 다룰게~

→ 즉, 제네릭은 모든 종류의 타입을 다룰 수 있도록, 클래스나 메서드를
타입 매개변수를 이용하여 선언한 기법

2. 제네릭(Generic)

제네릭 사용 방법?

: 클래스 명 뒤에 <E>, <K>, <V> 와 같은 형식으로 사용

단, < > 안에는 구현 시는 알파벳, 사용 시는, 클래스만 들어갈 수 있다.

제네릭 사용 구간

- 1) 제네릭 클래스: 클래스 뒤에 제네릭 기법 사용
- 2) 제네릭 메서드: 메서드 뒤에 제네릭 기법 사용

2. 제네릭(Generic)

제네릭타입 매개변수(관례적으로 널리 사용되는 타입명은 하나의 대문자 알파벳 사용)

- 1) E(Element) : 배열이나 집합들의 요소 타입을 지정
- 2) T(Type) : 변수의 자료형과 같은 타입을 지정
- 3) S,U(…) : 여러개의 제네릭 기법을 사용할 때 많이 사용
- 4) K(Key) : 키 타입 지정시 사용
- 5) V(Value) : 값 타입 지정 시 사용

2. 제네릭(Generic)

<실습> TypeSearch.java

제네릭 메서드를 사용하여, 들어온 값이 어떤 자료형인지 알려주는 실습

```
public class TypeSearch {  
    public static void main(String[] args) {  
        Value v= new Value();  
        v.typeSearch(3);  
        v.typeSearch(3.14);  
        v.typeSearch(3.14f);  
        v.typeSearch('a');  
        v.typeSearch("hello");  
    }  
}
```

<실행 결과>

```
3는 정수 입니다.  
3.14는 실수 입니다.  
3.14는 실수 입니다.  
a는 문자형 입니다.  
hello는 문자열 입니다.
```

```
class Value{  
    public <T> void typeSearch(T x){  
        if(x instanceof Integer) {  
            System.out.println(x+"는 정수 입니다.");  
        }  
        else if(x instanceof Double||x instanceof Float) {  
            System.out.println(x+ "는 실수 입니다.");  
        }  
        else if(x instanceof Character) {  
            System.out.println(x+"는 문자형 입니다.");  
        }  
        else if(x instanceof String) {  
            System.out.println(x+"는 문자열 입니다.");  
        }  
    }  
}
```


2. 제네릭(Generic)

<실습> Generic2.java

제네릭 클래스를 사용하여, 객체 생성시 멤버 타입을 지정하기

```
public class Generic2 {  
    public static void main(String[] args) {  
        A<Integer> a1 = new A<Integer>();  
        a1.x=3;  
        a1.y=10;  
        System.out.println(a1.x+a1.y);  
  
        A<String>a2 = new A<>();  
        a2.x="hello ";  
        a2.y="world!";  
        System.out.println(a2.x+a2.y);  
    }  
}  
  
class A<T>{  
    T x;  
    T y;  
}
```

<실행 결과>

```
13  
hello world!
```