Java Programming

컬렉션과 제네릭

2018. 12. 24

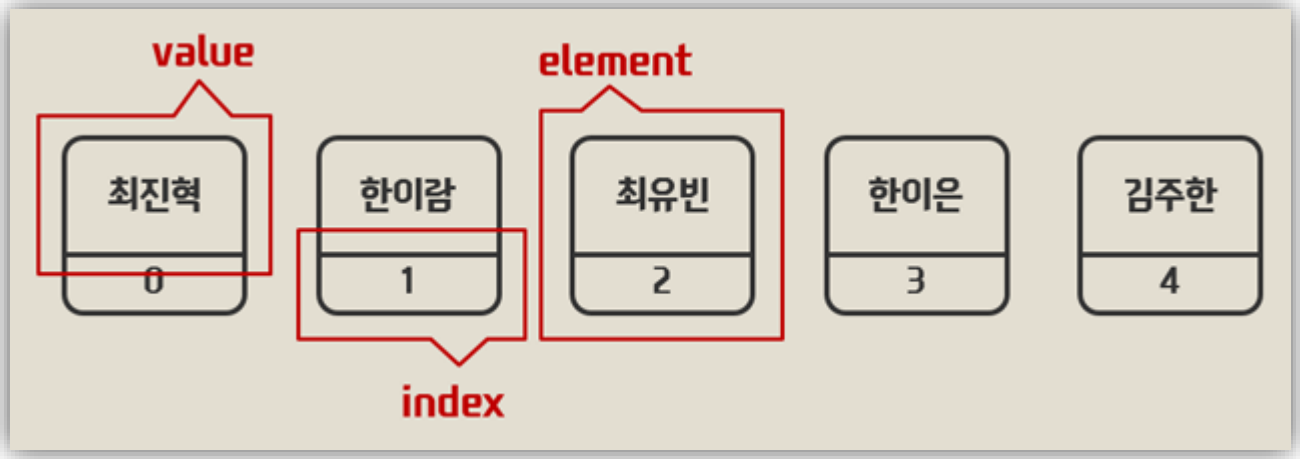
안광민

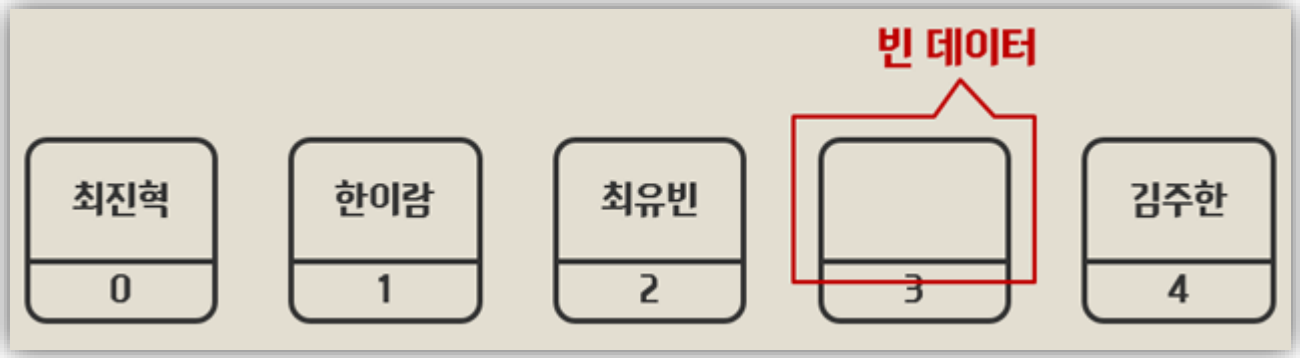
강의 진행 순서

1. 배열의 문제점
2. 컬렉션의 이해
3. 컬렉션의 종류
4. 제네릭의 이해
5. 제네릭 활용
6. 학습정리
7. 퀴즈

step
01

배열의 문제점





배열의 문제점

배열의 문제점 처리방법

컬렉션의 이해(1)

컬렉션(Collection)은 자료구조를 처리

- 자료구조란 자료를 저장할 수 있는 구조
- 다양한 형태의 데이터를 다루기 위해 일반화 기법을 사용
- 대표적인 컬렉션
  - List 는 순서가 있는 데이터를 다루는 자료구조
  - Map 은 Key 와 Value 구조를 가지는 자료구조
  - Set 은 중복을 허용하지 않는 자료구조

컬렉션의 이해(2)

컬렉션은 지원 도구

- 프로그래머가 데이터를 어떻게 저장하고 읽을 지가 아니라 **프로그램의 핵심기능**에 집중하도록 지원

컬렉션은 객체만 허용

- int, char, double 등의 **기본 타입 불가**

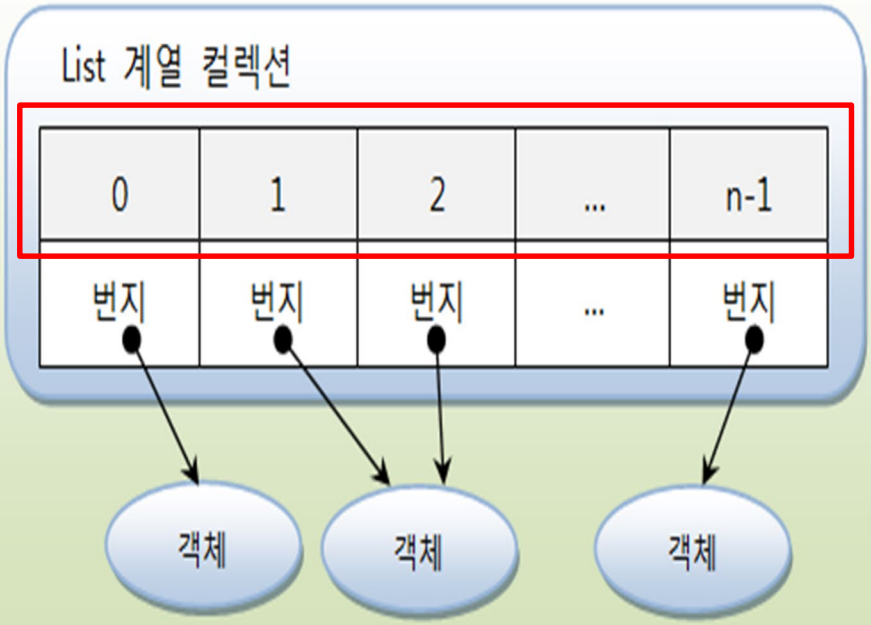
```
Vector<int> vi = new Vector<int>();                    // 컴파일 오류
```

수정

```
Vector<Integer> vi = new Vector<Integer>();            // 정상 코드
```

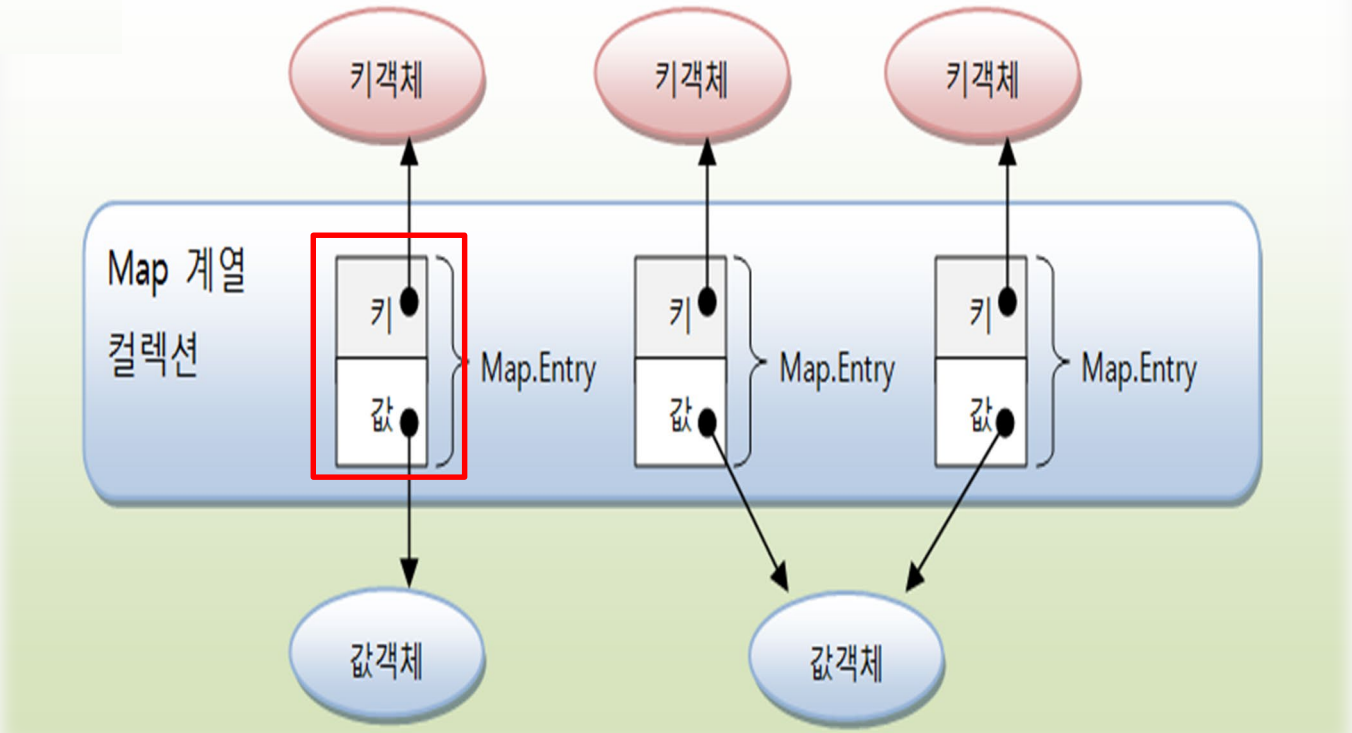
컬렉션의 종류

컬렉션의 종류 (LIST)



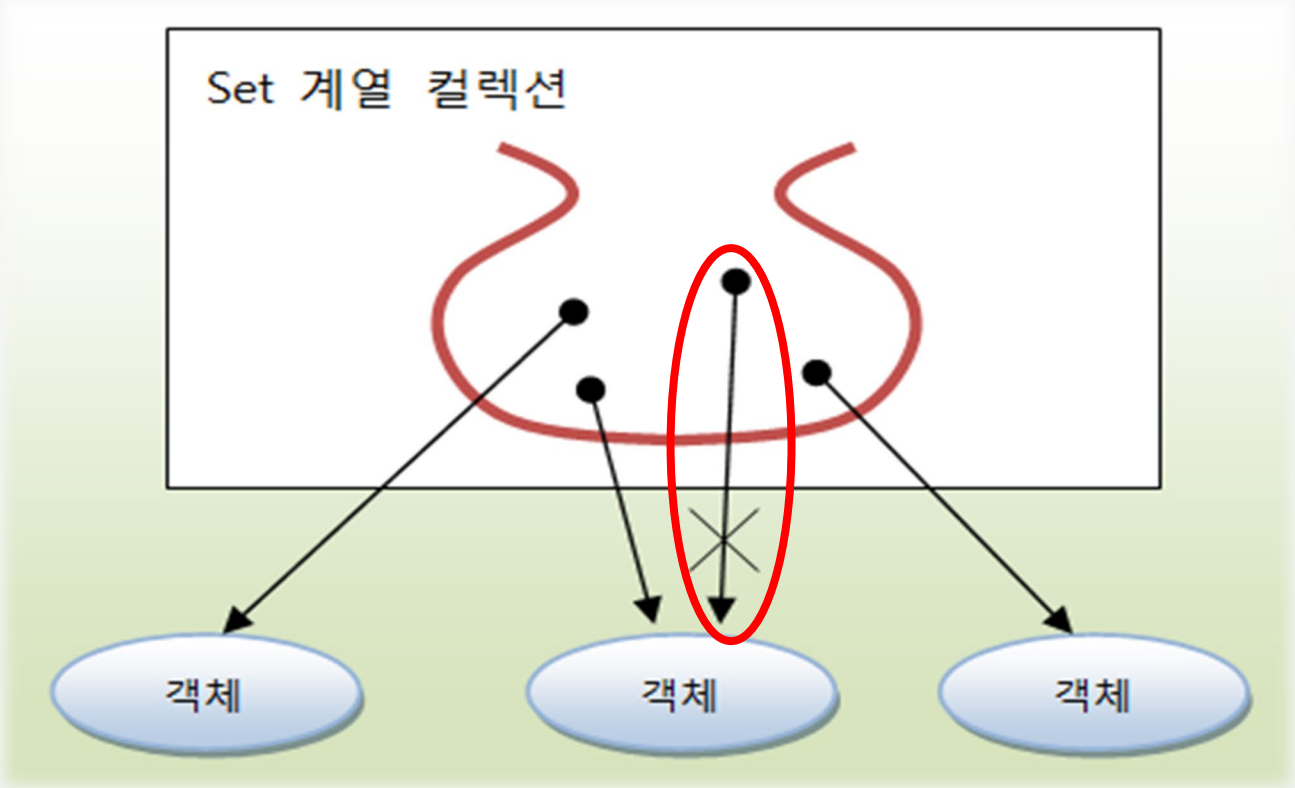
컬렉션의 종류

컬렉션의 종류 (MAP)



컬렉션의 종류

컬렉션의 종류 (SET)



제네릭의 이해

제네릭(Generic)

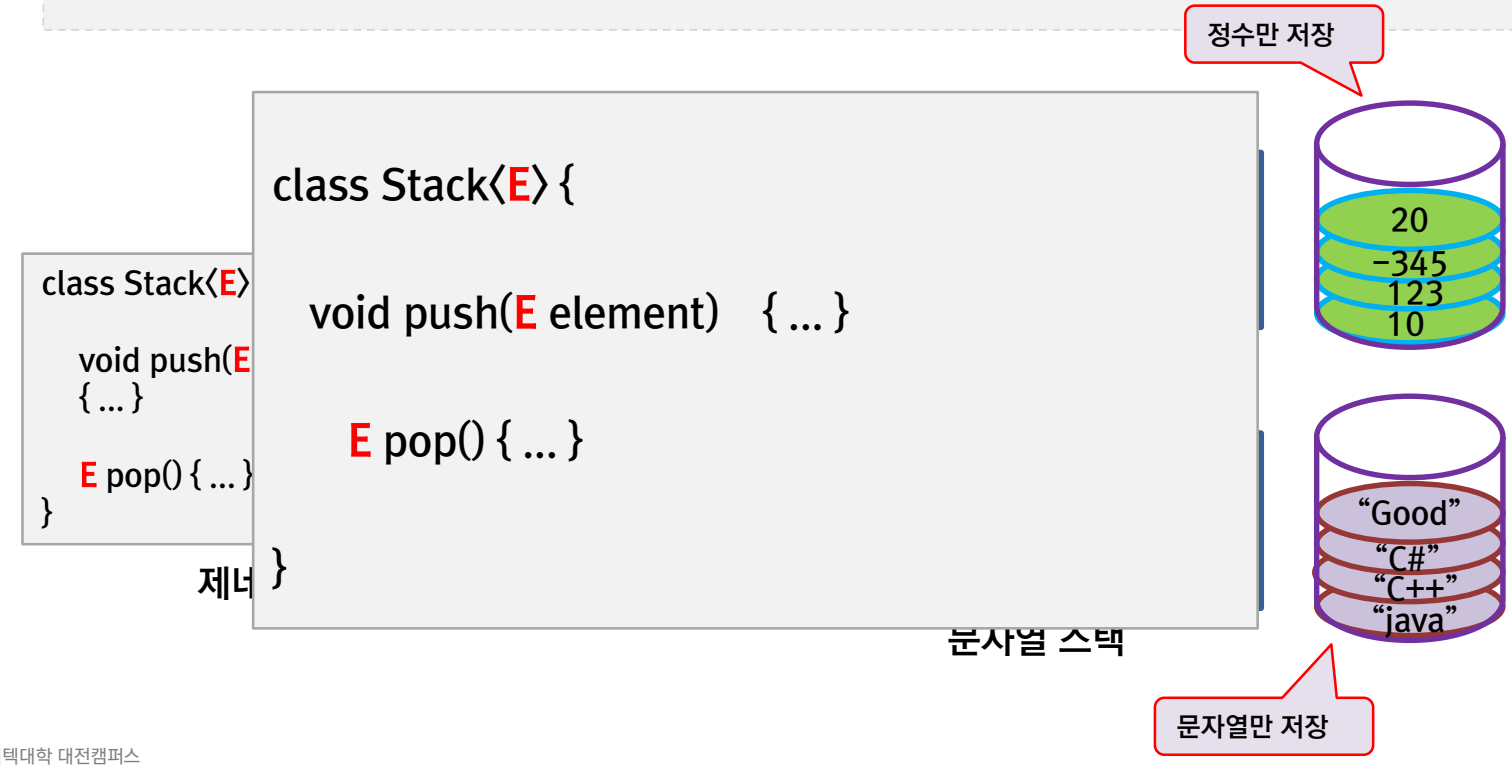
- 다양한 종류의 데이터 타입을 다룰 수 있도록 일반화
- 타입 매개 변수로 클래스,인터페이스, 메소드 작성

```
class Stack<E> {

   void push(E element)   { ... }

      E pop() { ... }

}
```



제네릭의 활용

제네릭 활용

- 클래스 이름 옆에 타입 매개 변수

```
public class MyClass<T> {
   T data;

   void set(T data) {
      this.data = data;
   }

   T get() {
      return data;
   }
}
```

타입 매개 변수 T (제네릭 클래스 선언)

data의 타입은 T

T 타입의 매개변수 선언

T 타입의 리턴값 선언

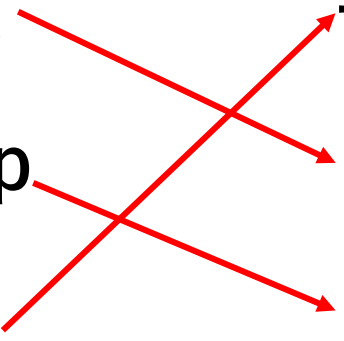
- 제네릭 객체 생성
  - 객체 생성 시에 타입을 지정하여 객체를 생성

```
MyClass<String> s = new MyClass<String>();

MyClass<Integer> n = new MyClass<Integer>();
```

학습정리

- 컬렉션은? 자료구조를 처리하는 클래스들
- 제네릭은? 다양한 구조를 처리하기 위해 일반화
- List
- Map
- Set



- 중복 되어서는 안 되는 데이터
- 순서가 있는 데이터
- Key-Value 형식의 데이터

퀴즈

퀴즈 1

- 학생정보관리 시스템을 개발할 때 적당한 컬렉션은?

① 리스트 (List)

해설: 학생정보는 학번 순서대로 처리하므로

② 트리셋 (TreeSet)

퀴즈 2

- 컴퓨터용어사전 프로그램을 개발할 때 적당한 컬렉션은?

① 맵 (Map)

해설: 용어사전은 키가 되는 용어와 값이 되는 설명이 필요하므로

② 연결리스트 (LinkedList)

퀴즈 3

- Set 컬렉션을 사용하면 좋은 프로그램은 어떤 것이 있을까요?

① 로또게임 (번호뽑기)

해설: 중복되는 번호가 있으면 안되므로

② 배틀그라운드 (사격게임)

묻고 답하기

감사합니다