

프로그래밍 언어 응용

chapter12

컬렉션 프레임워크

제공된 자료는 훈련생의 수업을 돕기 위한 것으로, 타인과 공유하시면 안됩니다.

Contents

part.1

제네릭

part.2

컬렉션 프레임워크

part.3

Collection 인터페이스

part.4

Map 인터페이스

제네릭이란?

- 제네릭은 클래스 내부에서 사용할 데이터타입을 외부에서 지정하는 기법이다.
- 제네릭을 사용하면 같은 클래스에서 여러 데이터 타입을 사용할 수 있다.

ArrayList 클래스의 구조

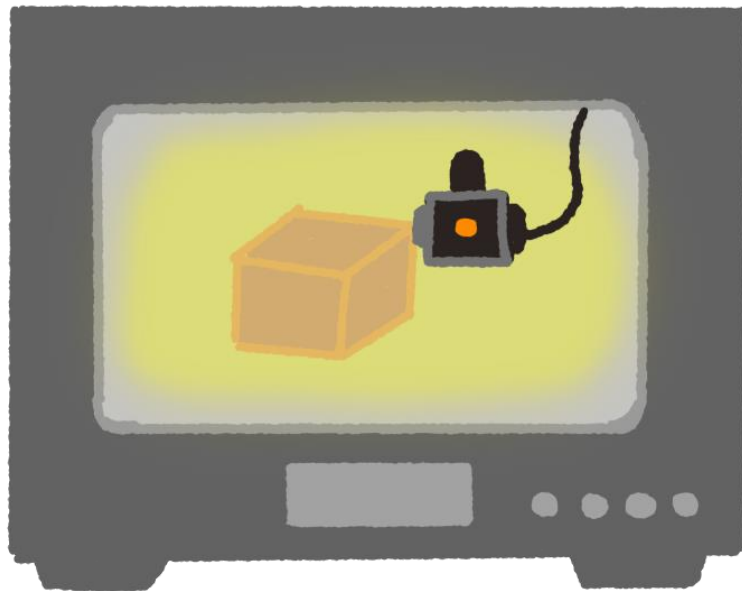
```
public class ArrayList<E> extends AbstractList<E>
```

ArrayList 생성하기

```
ArrayList<Integer> list1 = new ArrayList<>();  
ArrayList<String> list2 = new ArrayList<>();
```

3D 프린터 클래스를 만들어서 제네릭을 사용하는 예제를 작성한다.

3D 프린터는 파우더와 플라스틱을 재료로 사용하여 입체 모형을 만든다.

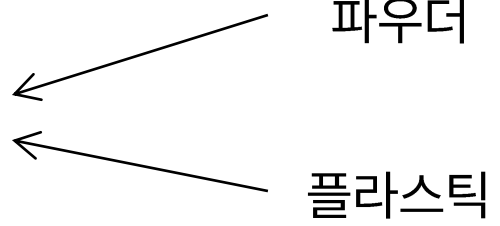


```
class ThreeDPrinter1 {  
    private Powder material;    //재료가 파우더인 경우  
}
```

```
class ThreeDPrinter2 {  
    private Plastic material;    //재료가 플라스틱인 경우  
}
```

그런데 재료만 다르고 프린터의 기능이 동일하다면 클래스를 통합하는 것이 효율적이다.
변수의 자료형을 'Object'로 사용하면 모든 종류의 재료를 저장할 수 있다.
아쉬운점은 값을 꺼낼 때, 원래타입으로 형 변환을 해주어야 한다.

```
class ThreeDPrinter {  
    private Object material;  
}
```



파우더

플라스틱

```
Powder powder = (Powder) printer.getMaterial();
```

제네릭 클래스를 만드는 방법

- 따라서 Object형을 사용하면 재료를 꺼낼 때 다시 형변환을 해야하기 때문에 번거롭다.
- '제네릭'을 사용하면 형변환없이 여러가지 자료형을 다룰 수 있다.
- 제네릭은 클래스를 선언할 때 자료형을 지정하지 않고, 클래스를 사용할 때 자료형을 지정하는 방식이다.
- 제네릭에서는 여러 자료형을 사용해야하는 부분에 Object 대신 문자로 표현한다.

```
class ThreeDPrinter<T> {  
    private T material;  
}
```

제네릭 자료형 or 타입변수

객체를 생성할 때는 실제로 사용할 자료형을 대입한다.
그러면 컴파일러가 소스코드를 번역할 때 문자T를 실제 자료형으로 변환한다.
따라서 데이터를 꺼낼 때마다 형변환을 할 필요가 없다.

```
ThreeDPrinter<Powder> printer = new ThreeDPrinter<>();
```

```
class ThreeDPrinter<T> {  
    private T material;  
    public void setMaterial(T material) {  
        this.material = material;  
    }  
    public T getMaterial() {  
        return material;  
    }  
}
```

소스코드

컴파일

```
class ThreeDPrinter {  
    private Powder material;  
    public void setMaterial(Powder powder) {  
        this.material = powder;  
    }  
    public Powder getMaterial() {  
        return material;  
    }  
}
```

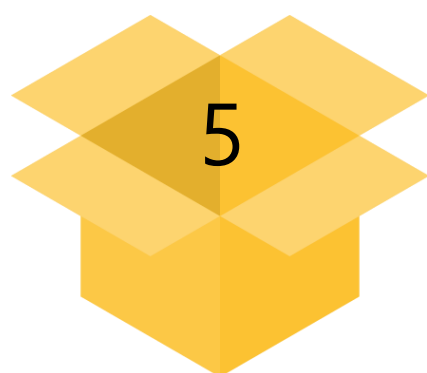
바이트코드

T가 Powder로 변경

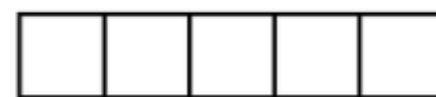
자료구조란?

- 프로그램에서 대량의 데이터를 저장할 때 사용한다.

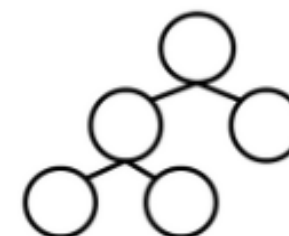
하나의 데이터 저장



여러 데이터 저장



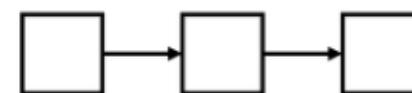
배열



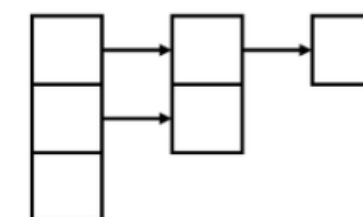
트리



스택



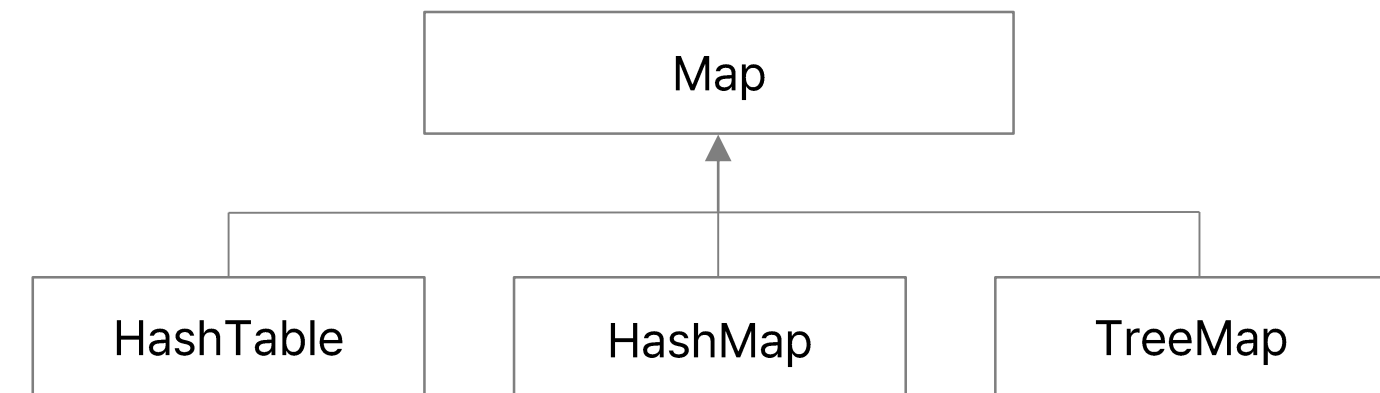
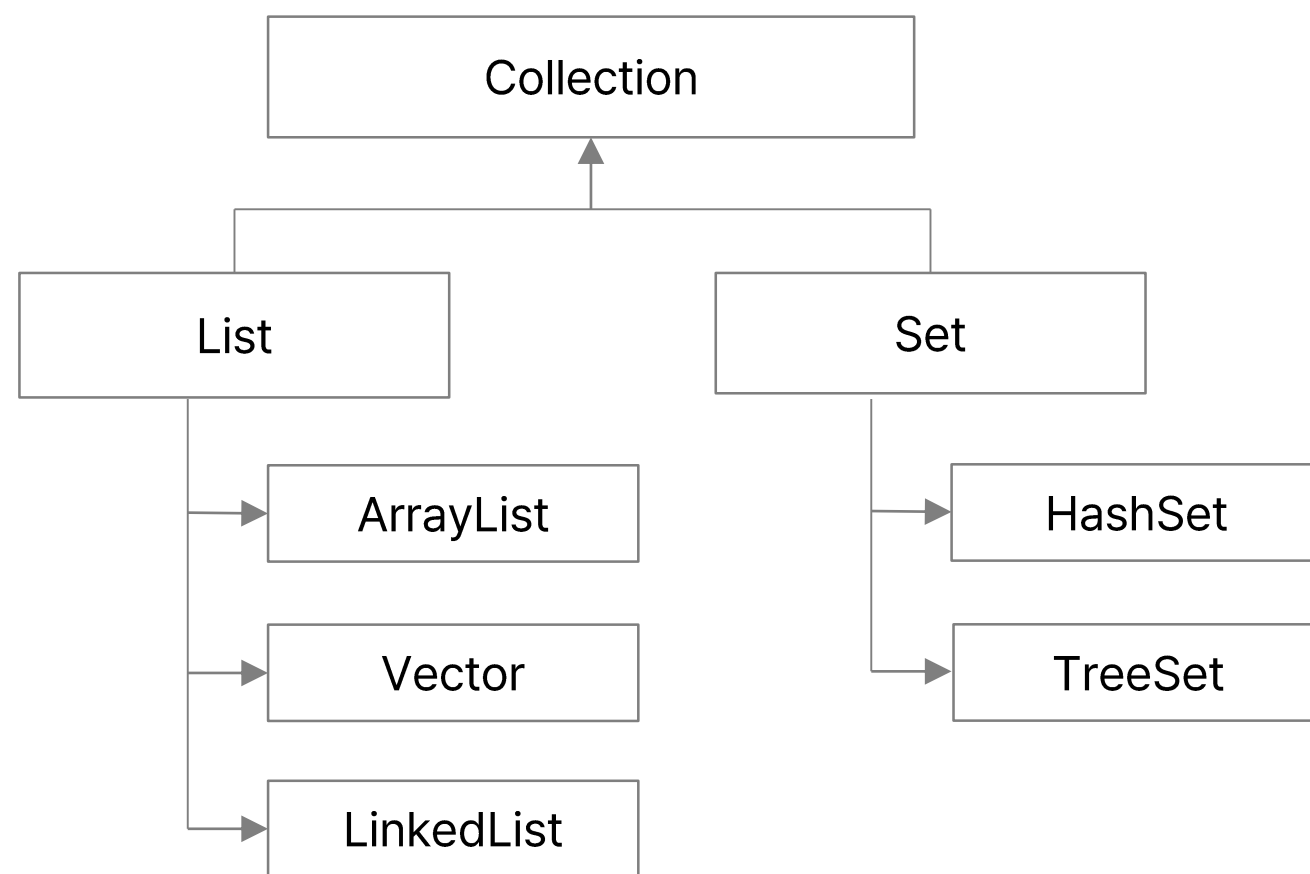
리스트



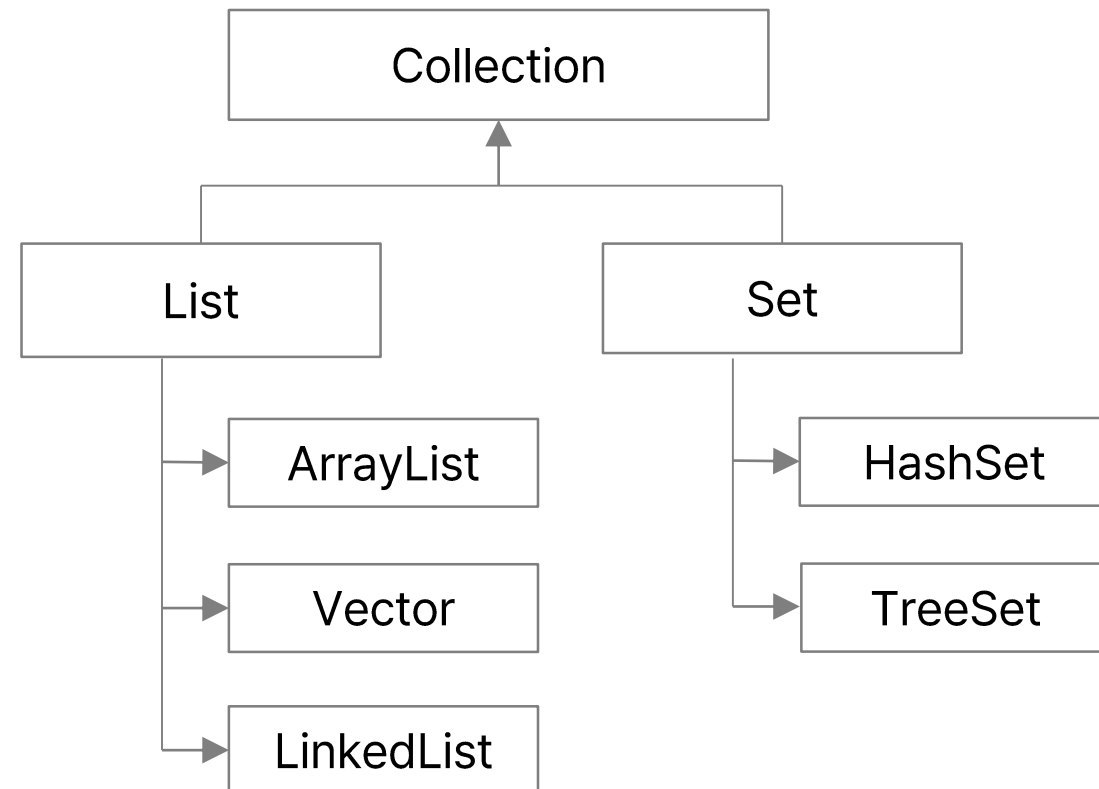
해시테이블

컬렉션 프레임워크란?

- 자바에서는 다양한 자료구조를 라이브러리로 제공하는데, 이를 컬렉션 프레임워크라고 한다.
- 크게 Collection과 Map 인터페이스로 이루어져 있다.
- 컬렉션 프레임워크에는 다양한 인터페이스와 구현 클래스가 있다.



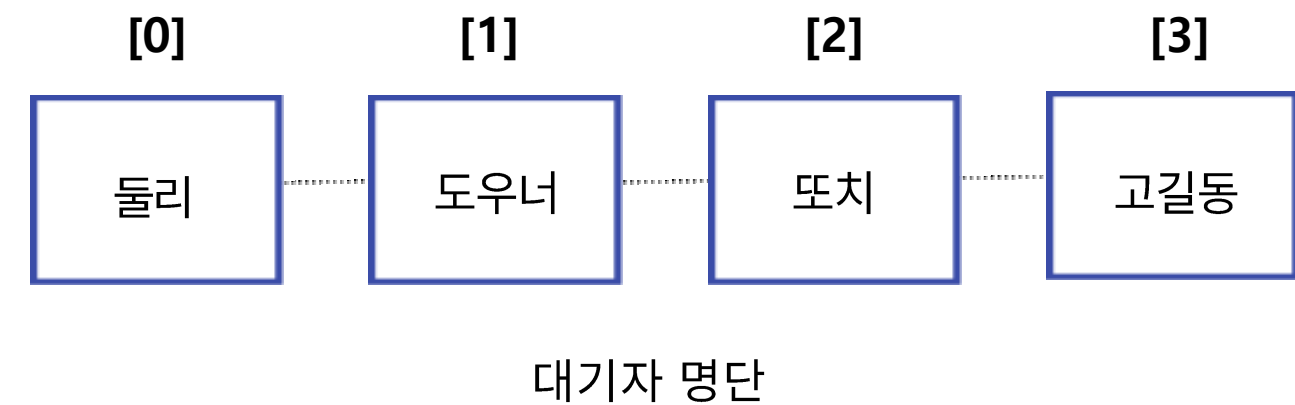
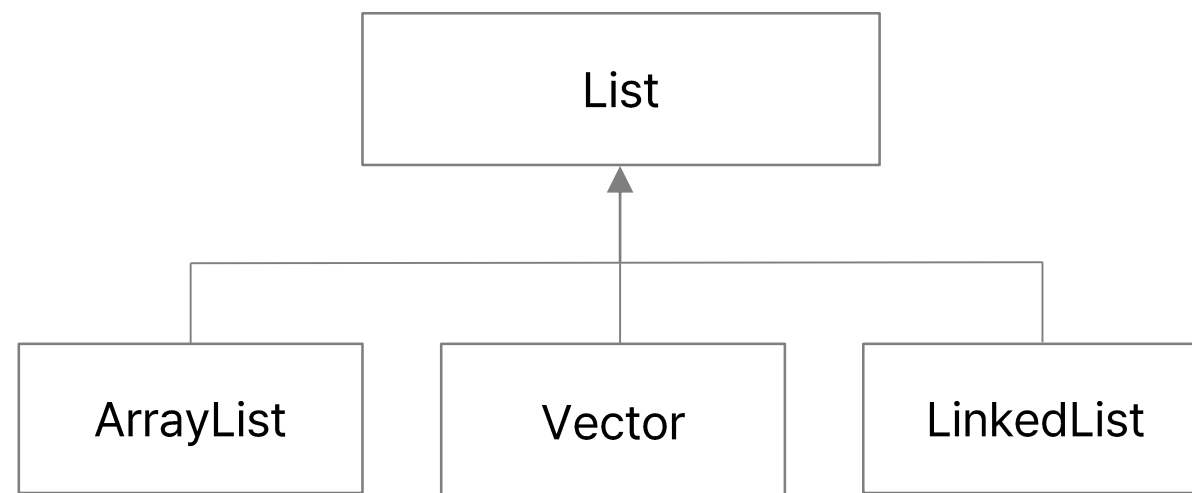
Collection 인터페이스는 하나의 자료를 모아서 관리하는데 필요한 기능을 제공한다.



분류	설명
List 인터페이스	<ul style="list-style-type: none">• 순서에 따라 저장• 중복 허용• 예) 대기자 명단
Set 인터페이스	<ul style="list-style-type: none">• 순서없이 저장• 중복을 허용하지 않음• 예) 회원 아이디

List란?

- List는 데이터를 순서에 따라 저장하고, 중복을 허용한다.
- ArrayList, Vector, LinkedList는 List의 특징을 구현한 클래스이다.



리스트에 저장된 모든 데이터를 삭제한다.

[0]	[1]	[2]
자바	스프링	엘라스틱

1. 첫번째 데이터부터 삭제하는 경우

`list.remove(0);` →

[0]	[1]
스프링	엘라스틱

`list.remove(1);` →

[0]
스프링

`list.remove(2);` → Error!

2. 마지막 데이터부터 삭제하는 경우

`list.remove(2);` →

[0]	[1]
자바	스프링

`list.remove(1);` →

[0]
자바

`list.remove(0);` → Clear!

Q. List을 활용하여 쇼핑몰의 회원을 관리하는 프로그램을 구현하세요.

MemberArrayList 클래스는 회원을 저장하기 위한 ArrayList을 멤버변수로 가지고 있습니다.

그리고 리스트에 회원을 추가하는 메소드, 리스트에서 회원을 삭제하는 메소드, 회원 리스트를 출력하는 메소드를 가지고 있습니다.

MemberArrayList 클래스를 사용하여 아래와 같이 회원을 추가하고 삭제하세요.

MemberArrayList 클래스

```
ArrayList<Member> list;
```

```
void addMember(Member member)
```

```
void removeMember(int memberId)
```

```
void showAllMember( )
```

테스트 결과

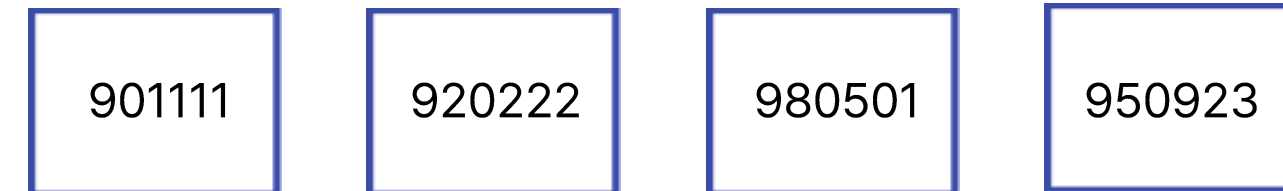
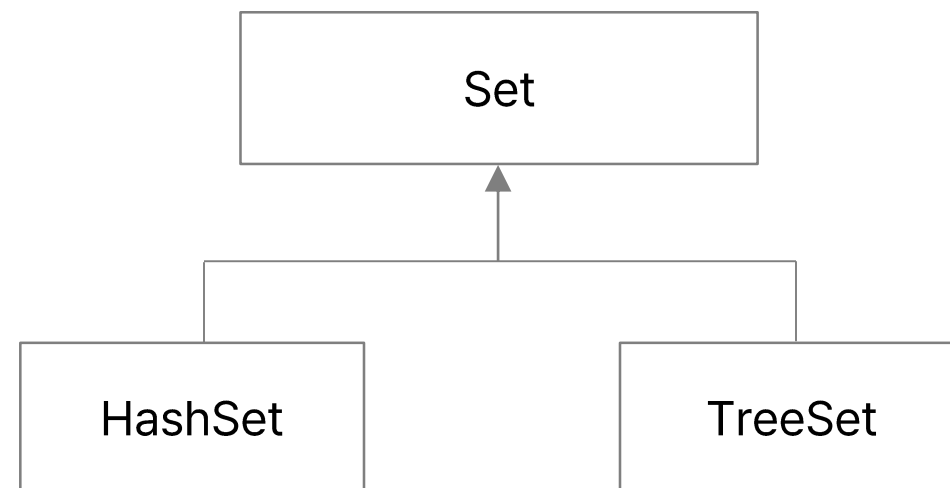
```
Problems Javadoc Declaration Console × Terminal History Coverage
<terminated> Quiz4 (50) [Java Application] C:\Users\Himjiyeon\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20
MemberArrayList를 생성하고 4명의 회원을 추가하세요!
Member [memberId=1001, memeberName=둘리]
Member [memberId=1002, memeberName=또치]
Member [memberId=1003, memeberName=도우너]
Member [memberId=1004, memeberName=고길동]

MemberArrayList에 1004번 회원이 있으면 삭제하세요!
1004번 회원을 삭제하였습니다
Member [memberId=1001, memeberName=둘리]
Member [memberId=1002, memeberName=또치]
Member [memberId=1003, memeberName=도우너]
```

Collection 인터페이스 Set

Set이란?

- Set은 데이터를 추가할 때 순서가 없고, 중복을 허용하지 않는다.
- HashSet, TreeSet은 Set의 특징을 구현한 클래스이다.



주민번호

Set은 인덱스가 없기 때문에 get() 메소드를 사용하여 데이터를 찾을 수 없다.

그래서 Set클래스는 데이터를 순차적으로 접근할 수 있는 iterator를 제공한다.

iterator를 사용하면 Set의 요소를 인덱스 없이도 접근할 수 있다.

Set에서 Iterator 객체를 생성하고, Iterator 클래스를 사용하여 데이터를 꺼낼 수 있다.

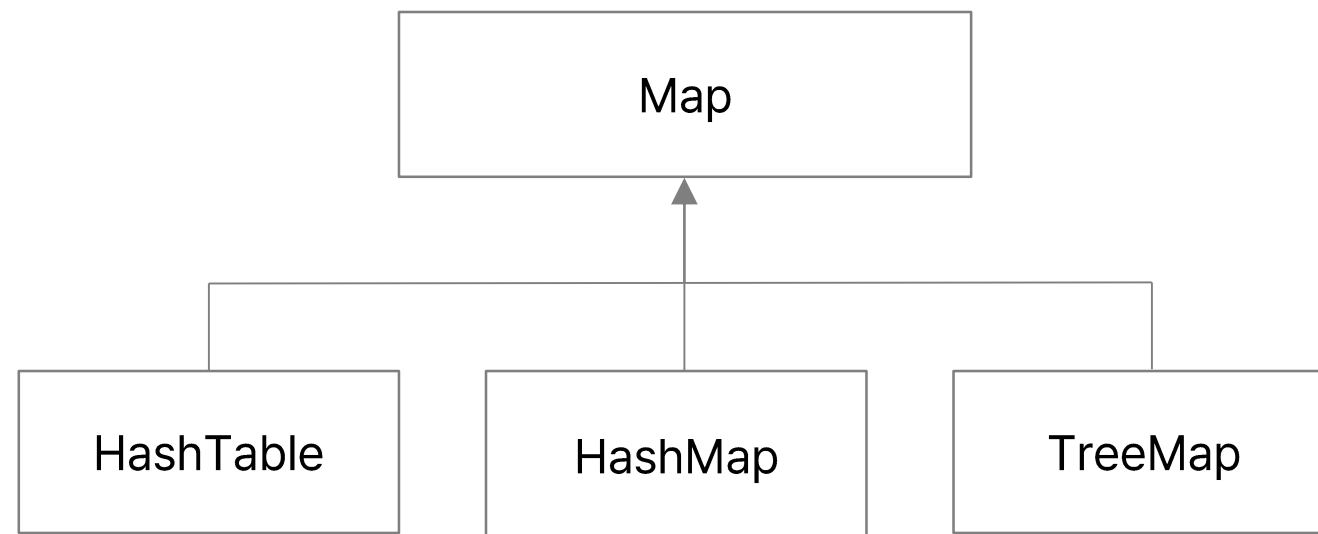
```
Iterator iterator = hashSet.iterator();  
int value = iterator.next();
```

Iterator 메소드	설명
boolean hasNext()	다음 요소가 있는지 확인한다.
E next()	다음 요소를 반환한다.

컬렉션 메소드	설명
boolean add(Object o)	컬렉션에 데이터를 추가한다.
boolean add(int index, Object o)	특정 위치에 데이터를 추가한다
void clear()	모든 요소를 삭제한다.
boolean remove (Object o)	데이터가 존재하면 삭제한다.
Object remove (int index)	특정 위치에 있는 데이터를 삭제한다.
int indexOf(Object o)	데이터의 위치를 찾는다.
Object get(int index)	특정 위치에 있는 데이터를 꺼낸다.
Object[] toArray()	컬렉션 안에 있는 배열을 꺼낸다.
int size ()	컬렉션의 크기를 반환한다.
Iterator<E> iterator	컬렉션을 순환할 반복자(iterator)를 반환한다.

Map이란?

- Map 인터페이스는 쌍으로 된 자료들을 관리하는데 필요한 기능을 제공한다.
- Hashtable, HashMap, TreeMap은 map의 특징을 구현한 클래스이다.



Map의 특징

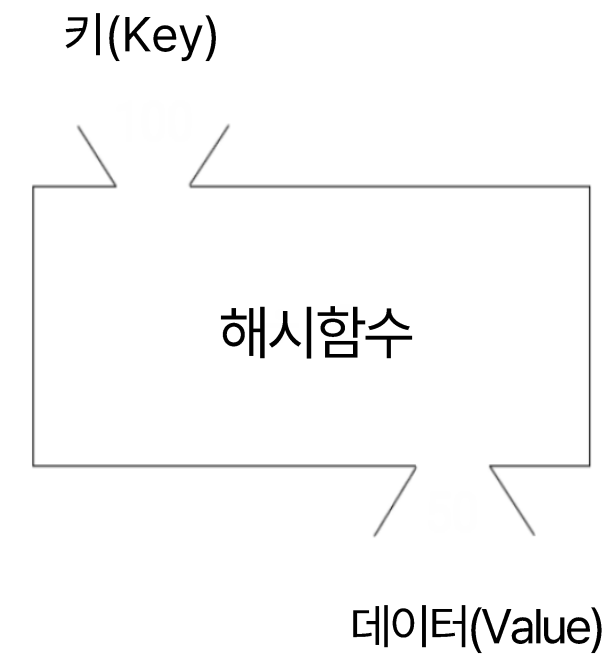
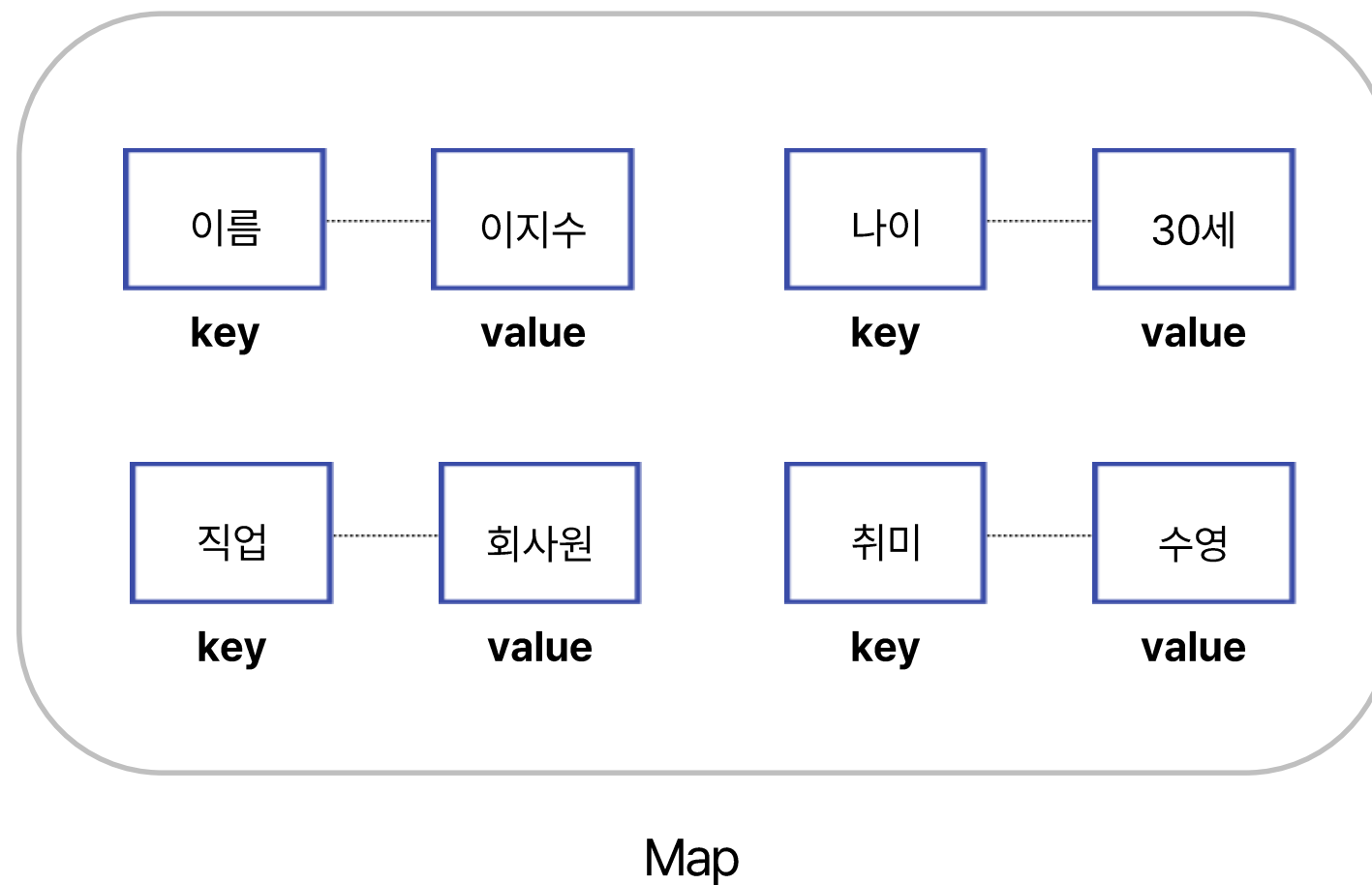
- Map은 데이터를 키(key)와 값(value)의 쌍으로 저장한다.
- 다음과 같이 사람의 정보를 저장한다면 "이름, 나이" 등은 key가 되고, "이지수, 30세" 등은 value가 된다.
- 키 값은 중복될 수 없다.

Key	Value
이름	이지수
나이	30세
직업	회사원
취미	수영

Key - 중복 X
Value - 중복 O

Map은 키와 해시함수를 사용하여 데이터를 찾는다.

해시함수를 사용하여 데이터의 위치를 찾고, 해당위치에서 데이터를 반환한다.



Q. Map을 활용하여 쇼핑몰의 회원을 관리하는 프로그램을 구현하세요.

MemberHashMap 클래스는 회원을 저장하기 위한 HashMap을 멤버변수로 가지고 있습니다.
그리고 맵에 회원을 추가하는 메소드, 맵에서 회원을 삭제하는 메소드, 회원 목록을 출력하는 메소드를 가지고 있습니다.

MemberHashMap 클래스를 사용하여 아래와 같이 회원을 추가하고 삭제하세요.

MemberHashMap 클래스

```
HashMap<Integer, Member> map;
```

```
void addMember(Member member)
```

```
void removeMember(int memberId)
```

```
void showAllMember( )
```

테스트 결과

```
Problems @ Javadoc Declaration Console X Synchronize Packages
<terminated> Quiz5 (9) [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (2024. 5. 7. 오전 10:08:09 - 오전 10:08:09) [pid: 8886]
MemberHashMap를 생성하고 4명의 회원을 추가하세요!
Member [memberId=1001, memeberName=둘리]
Member [memberId=1002, memeberName=또치]
Member [memberId=1003, memeberName=도우너]
Member [memberId=1004, memeberName=고길동]

MemberHashMap에 1004번 회원이 있으면 삭제하세요!
1004번 회원을 삭제하였습니다
Member [memberId=1001, memeberName=둘리]
Member [memberId=1002, memeberName=또치]
Member [memberId=1003, memeberName=도우너]
```

Map 인터페이스

Map의 메소드

Map 메소드	설명
V put (K key, V value)	Map에 데이터를 추가한다.
V remove(Object key)	키에 일치하는 데이터를 삭제한다.
V get(K key)	키에 일치하는 데이터를 찾는다.
boolean containsKey(Object key)	Map에 해당 키가 있는지 확인한다.
boolean containsValue(Object value)	Map에 해당 데이터가 있는지 확인한다.
Set keySet()	모든 key 값을 Set 객체로 반환한다. Key는 중복될 수 없으므로 반환형이 Set이다.
Collection values()	값을 collection으로 반환한다. Value는 중복이 가능하므로 반환형이 Collection이다.