

클릭하세요 자바

최문환

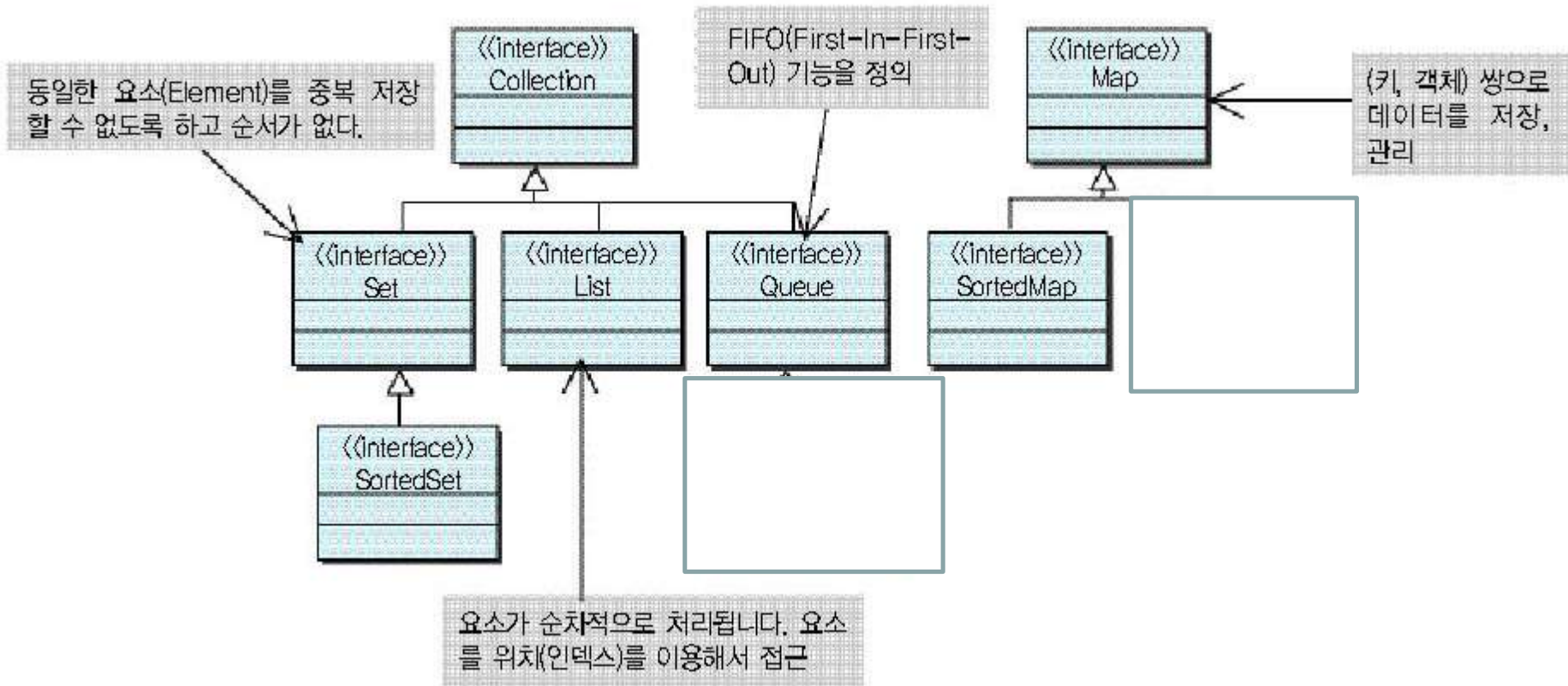


24장 컬렉션과 제네릭

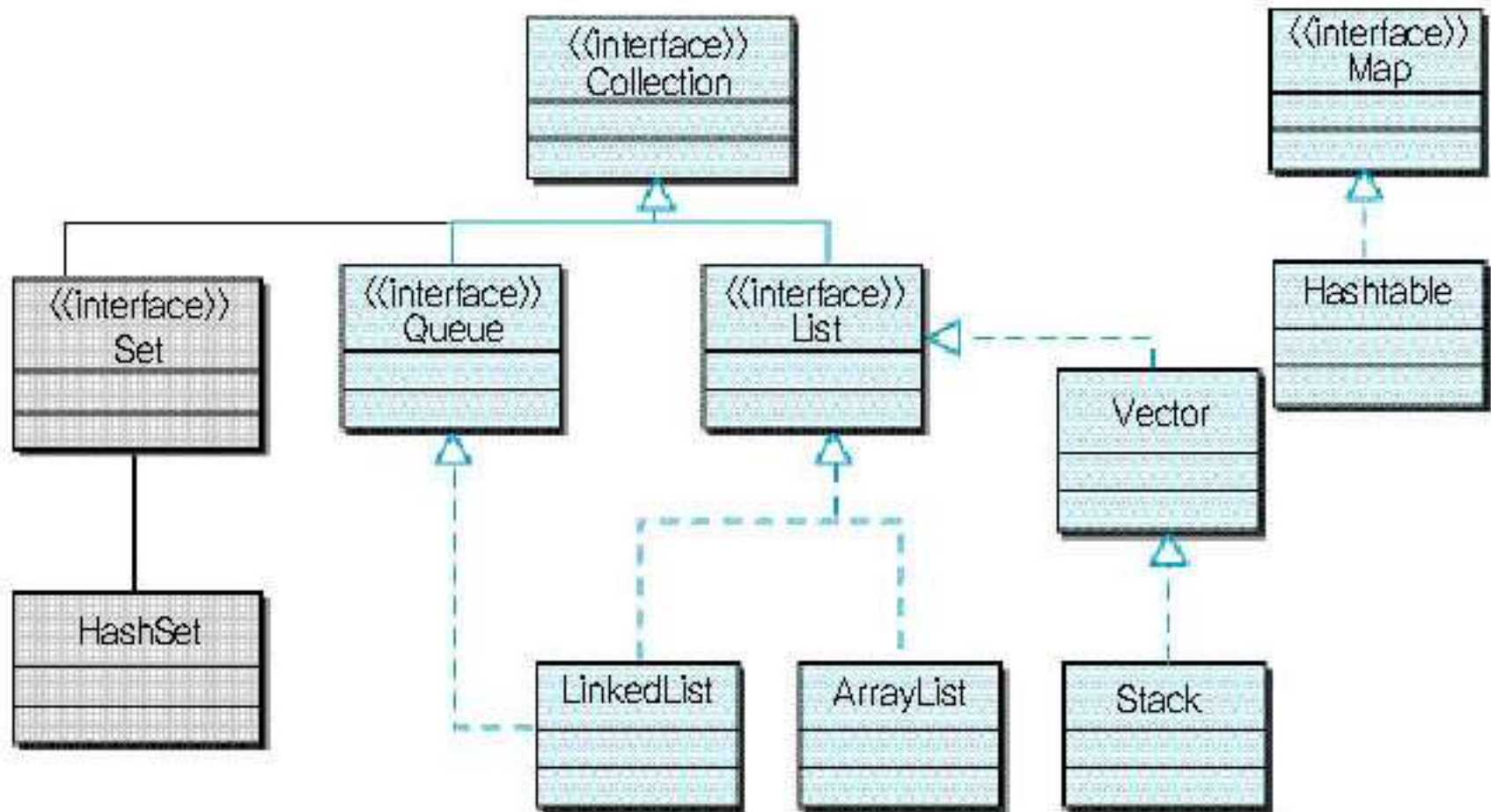
1. 컬렉션
2. Set 인터페이스
3. List 인터페이스
4. Stack 클래스
5. Queue 인터페이스와 LinkedList 클래스
6. Map 인터페이스와 Hashtable 클래스
7. 제네릭이란?



1. 컬렉션



구현 컬렉션 계층도



- Set 인터페이스

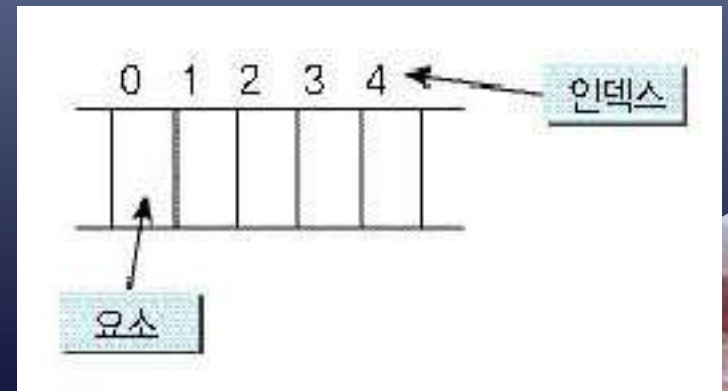


List 인터페이스



List 인터페이스

메서드	설명
<code>int indexOf(Object o)</code>	전달인자로 준 객체를 앞에서부터 찾아 해당 위치를 반환. 못 찾으면 -1 반환
<code>int lastIndexOf(Object o)</code>	객체를 마지막 위치부터 찾음
<code>E get(int index)</code>	index 위치의 객체를 반환
<code>E set(int index, E element)</code>	index 위치의 요소를 element로 주어진 객체를 대체한다.
<code>void add(int index, E element)</code>	index 위치에 element로 주어진 객체를 저장한다. 해당 위치의 객체는 뒤로 밀려난다.
<code>E remove(int index)</code>	index 위치의 객체를 지운다.



ArrayList 클래스



Iterator 인터페이스



Enumeration 인터페이스



Vector 클래스



Vector 클래스



<예제> 벡터 이용하기



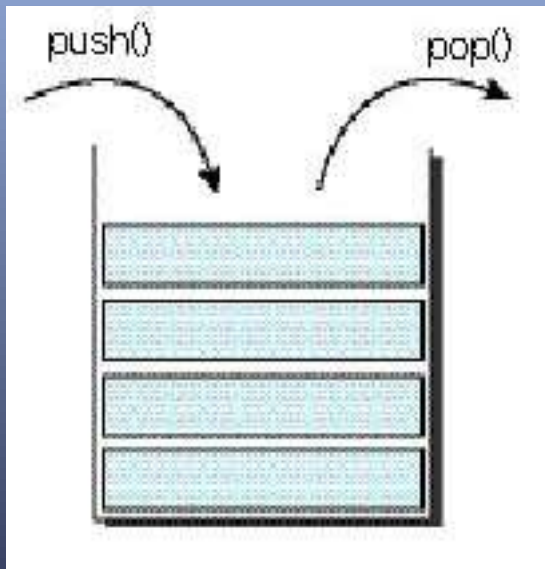
<예제> 벡터 요소 검색과 삭제



<예제> 벡터 요소 검색과 삭제



Stack 클래스



Stack 클래스

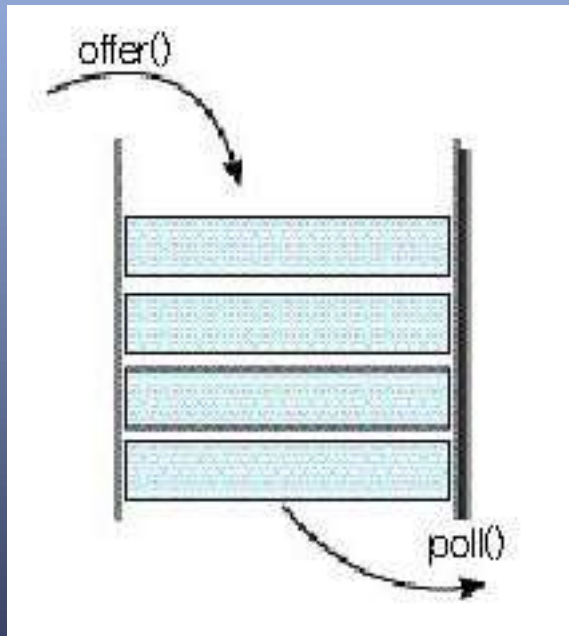




<예제> LIFO 구조의 스택 다루기



Queue 인터페이스와 LinkedList 클래스



Queue 인터페이스와 LinkedList 클래스



<예제> FIFO 구조의 큐 다루기



Map 인터페이스와 Hashtable 클래스

<예제> 해쉬 테이블 다루기



제네릭이 필요하게 된 배경



제네릭이 필요하게 된 배경



제네릭 타입



확장 for 문



제네릭 타입



제네릭 타입



제네릭이란?



<예제> Object형으로 클래스 설계



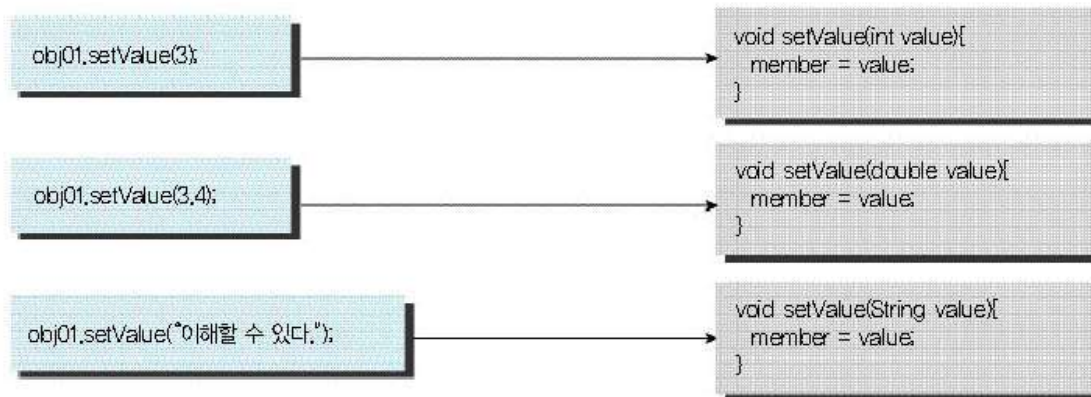
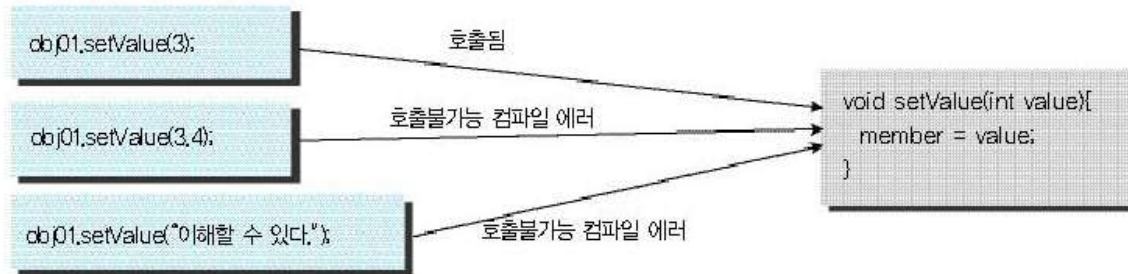
<예제> Object형으로 설계한 클래스의 단점



제네릭이란?



제네릭이란?



obj01.setValue(3);

obj01.setValue(3.4);

obj01.setValue("이해할 수 있다.");

void setValue(T value){
 member = value;
}

void setValue(int value){
 member = value;
}

void setValue(double value){
 member = value;
}

void setValue(String value){
 member = value;
}

