

Data Engineering

데이터엔지니어링

[리스트 기반 데이터엔지니어링(1)]



Java Collection Framework와 ArrayList 소개



학습내용

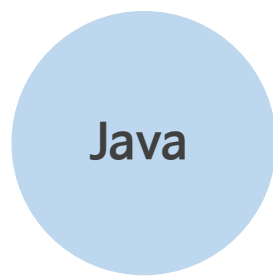
- 1 Java Collection Framework
- 2 ArrayList

학습목표

- Java Collection Framework의 개념을 설명할 수 있다.
- ArrayList의 개념을 설명할 수 있다.

Java Collection Framework





데이터의 모음

일반적인 기능이
추상화된 소프트웨어

Java Collection Framework

효율적인 데이터 접근 조작 가능

조직

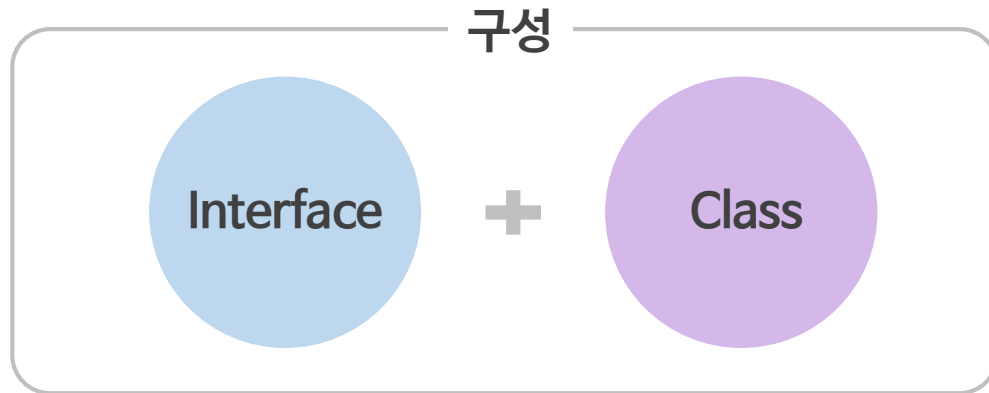
관리

저장



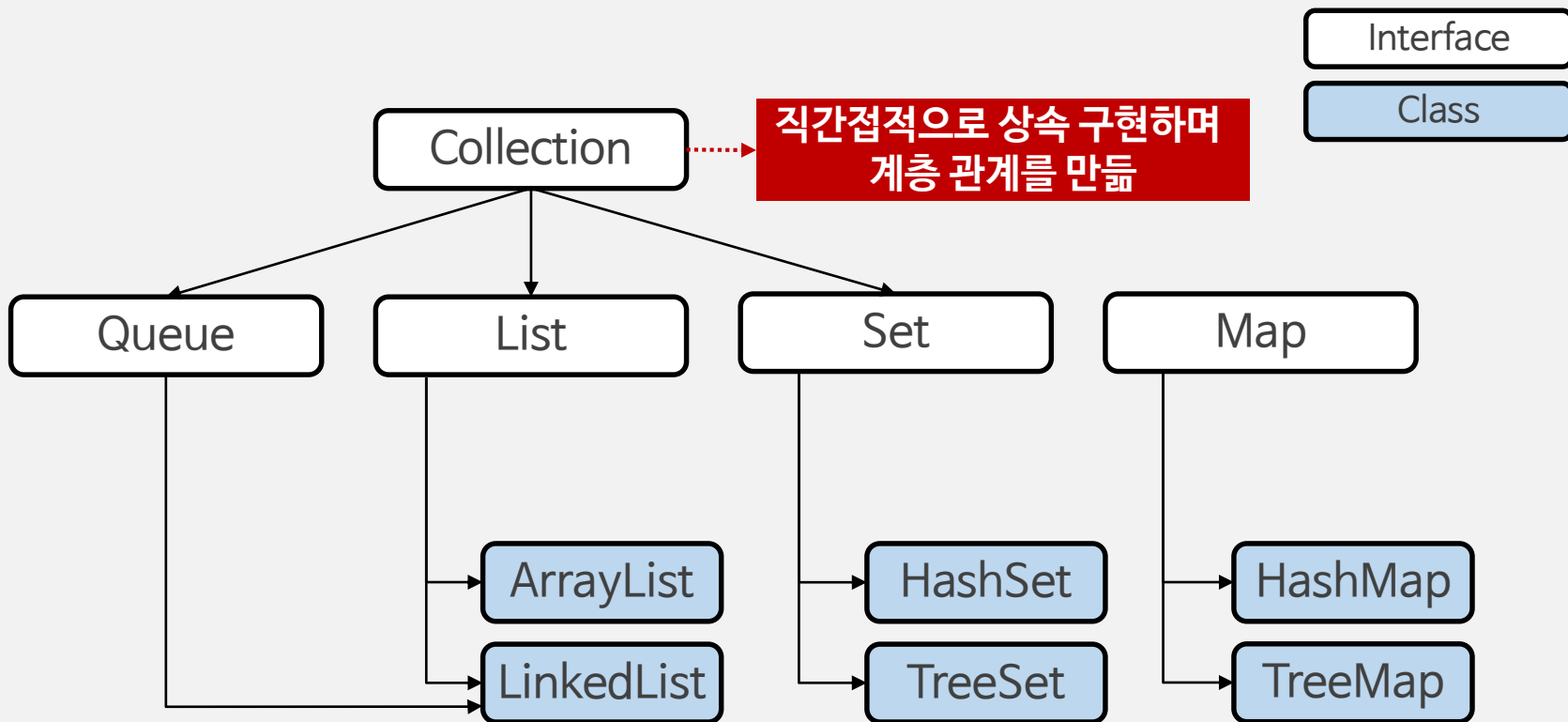
추상화

Java Collection Framework



JDK 1.2 버전부터 제공

계층 구조

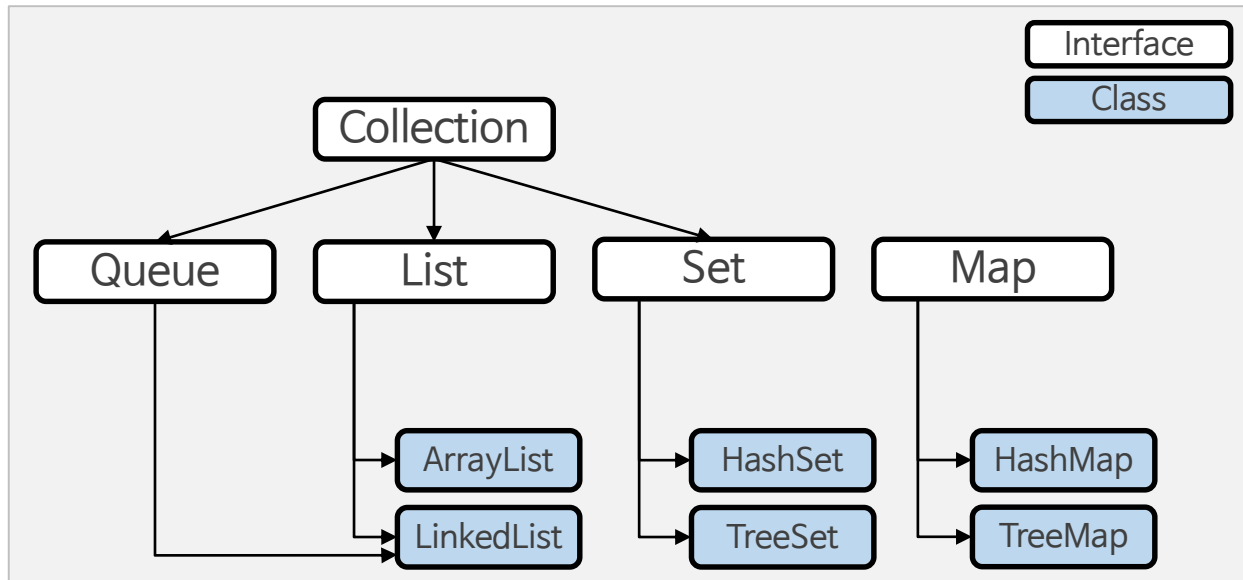


Collection

Java Collection Framework의 **최상위 Interface**

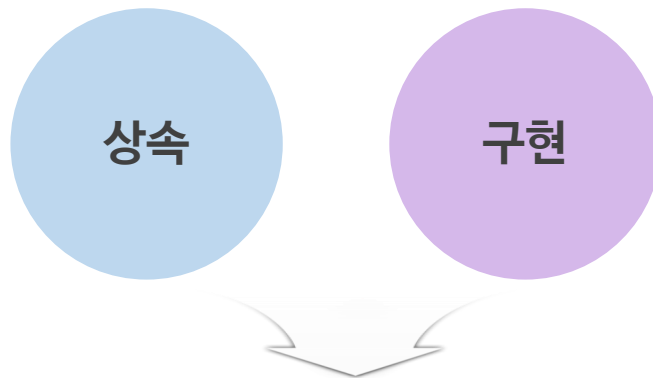
<https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/util/Collection.html>

Collection



Collection에 대한 일반적인 연산

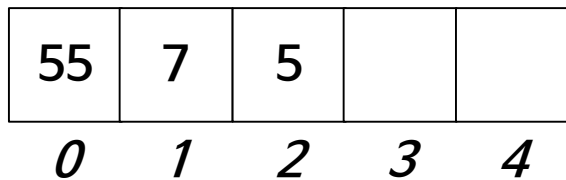
Collection Interface



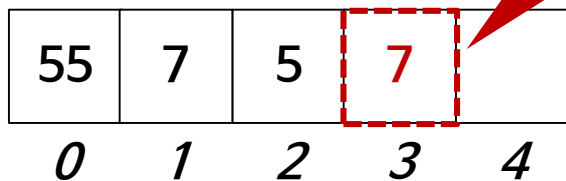
다양한 성질

중복 측면

ArrayList

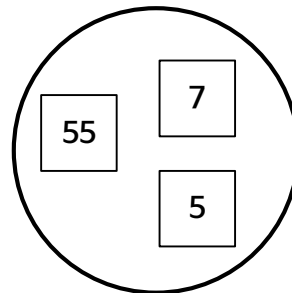


add 7

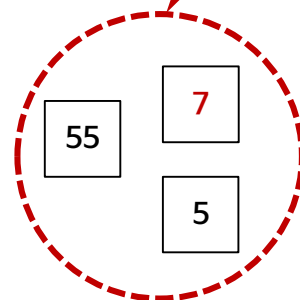


중복 허용

HashSet



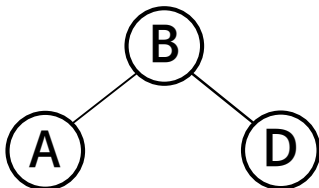
add 7 →



중복 허용 X

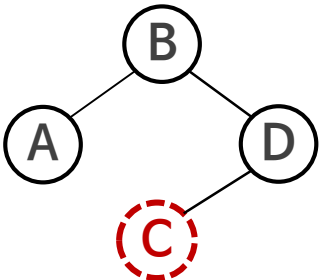
순서 측면

TreeSet

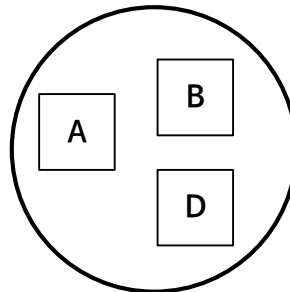


순서 유지

add C

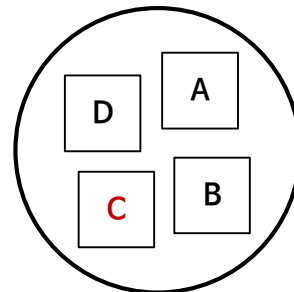


HashSet



순서 없음

add C



자료 입력 측면

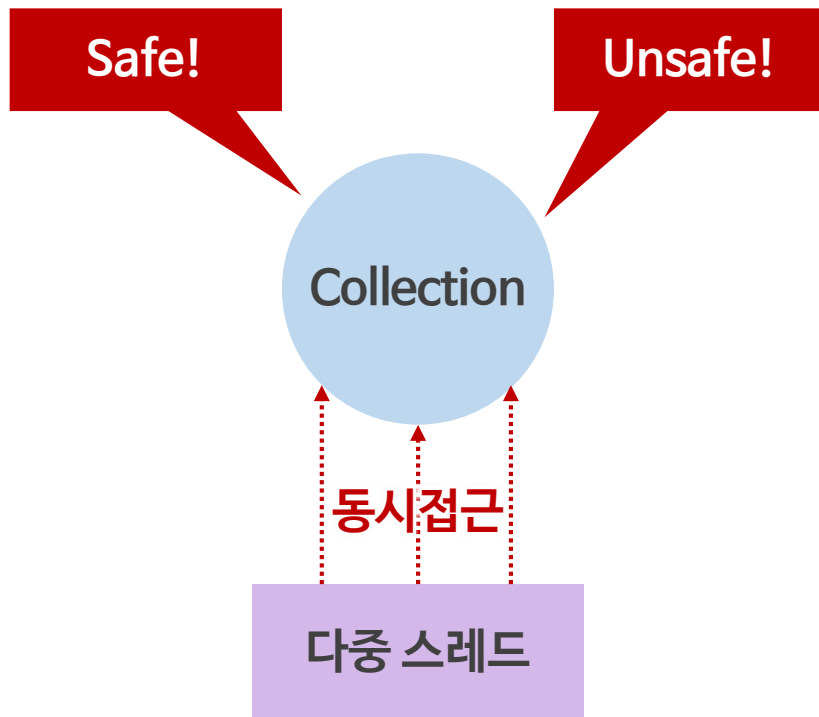
ArrayList

자료가 입력되는 순서를
Index를 통해 유지

HashSet

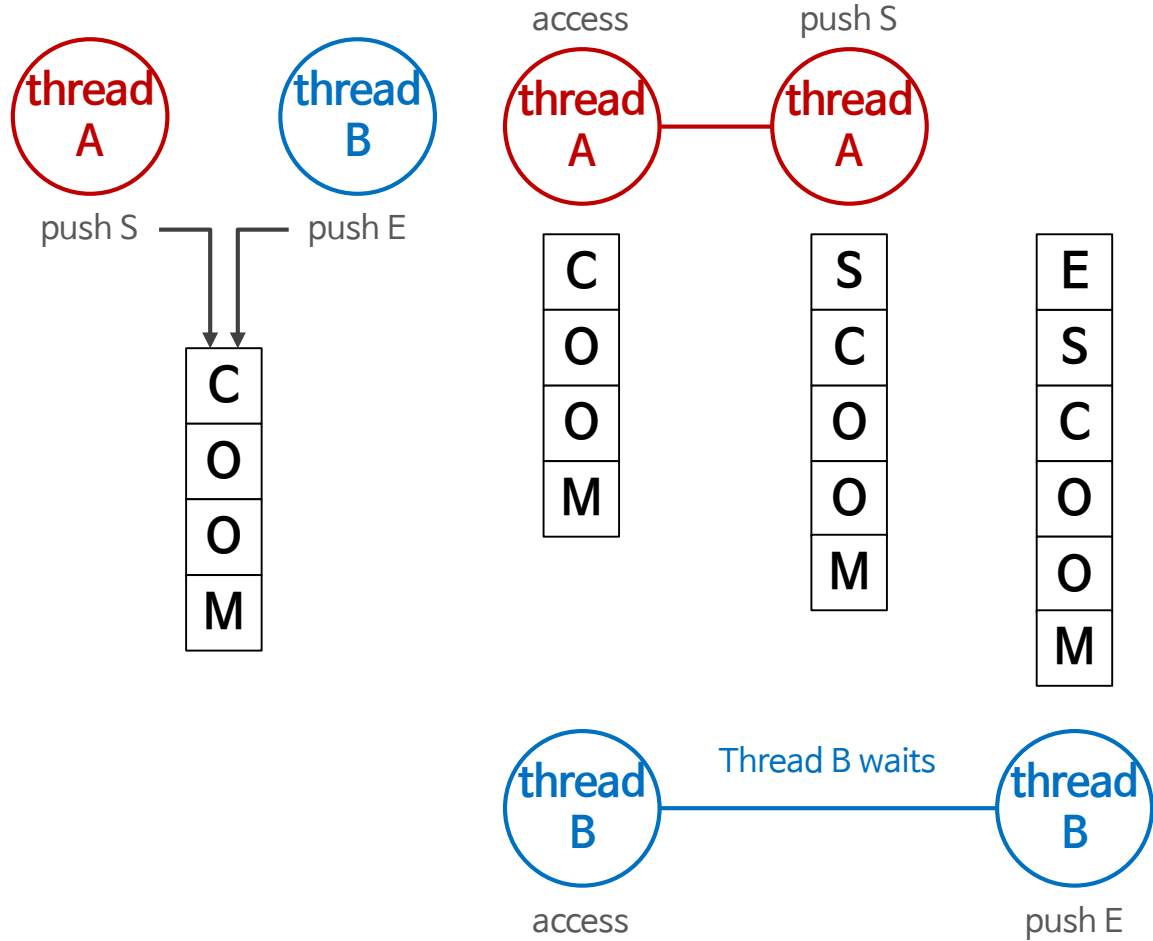
자료 입력의 순서도 사라짐

멀티스레드 측면



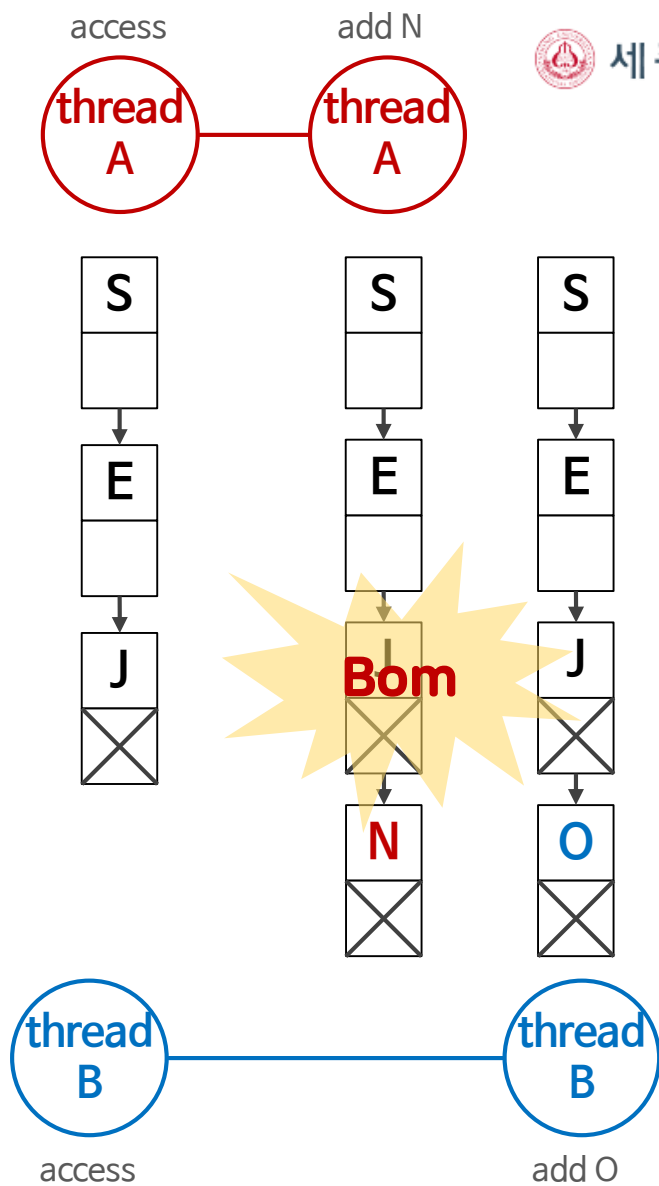
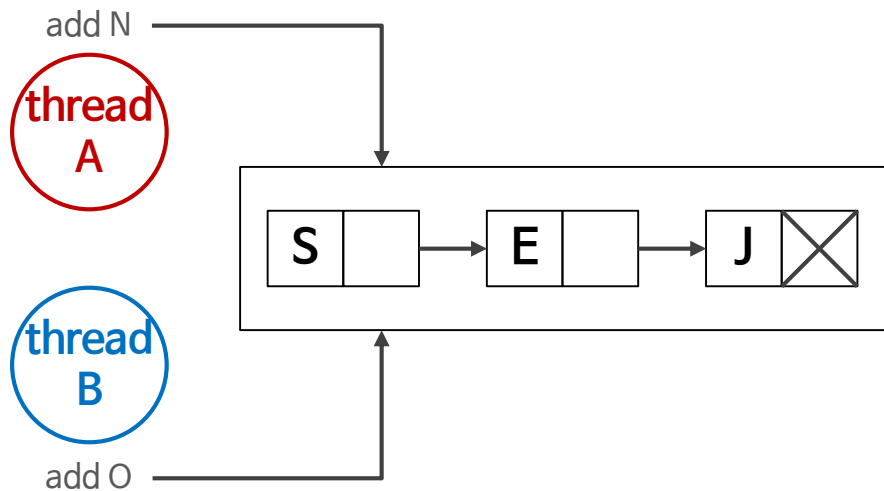
멀티스레드 측면

Stack

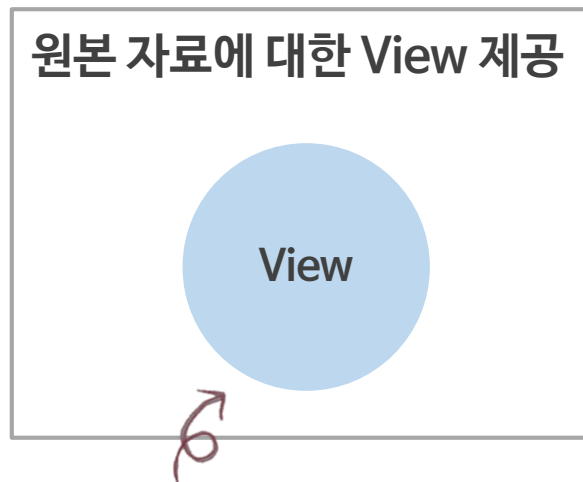


멀티스레드 측면

LinkedList



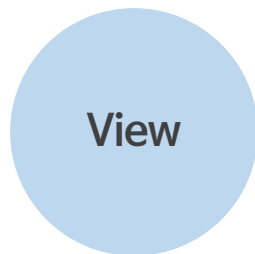
View Collections



원본 Collection에 대한 접근

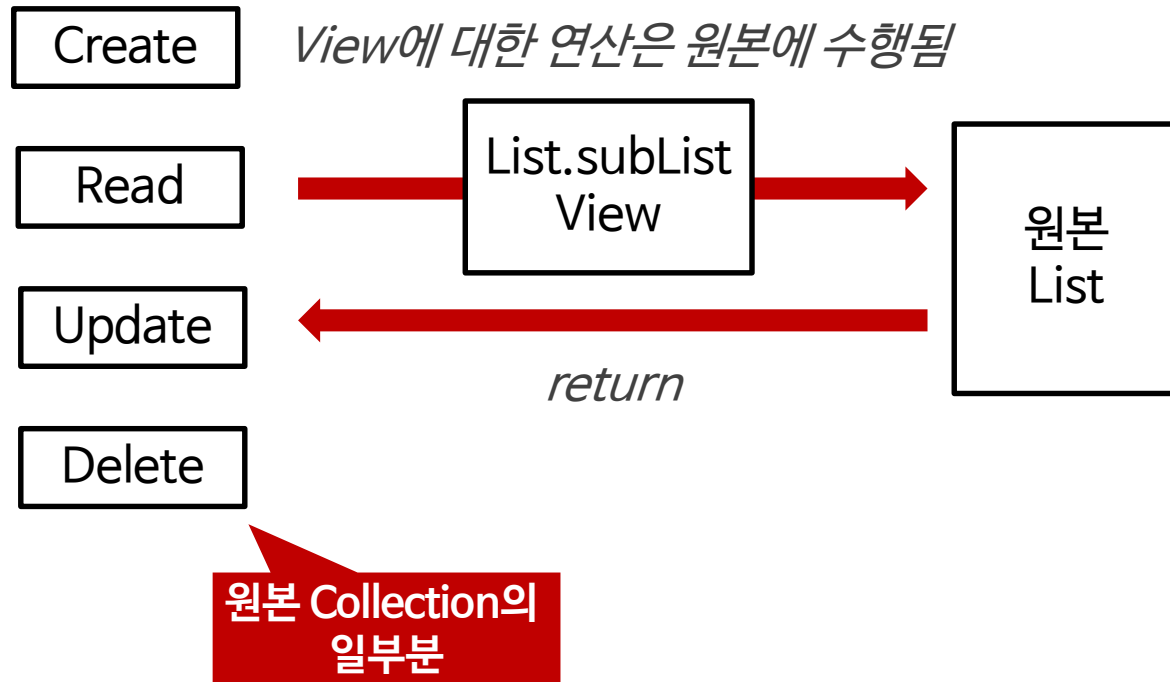
View Collections

원본 자료에 대한 View 제공

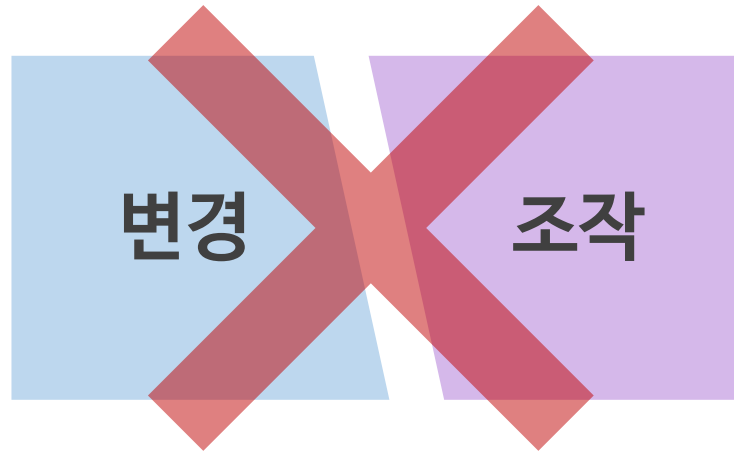


동기화 용이

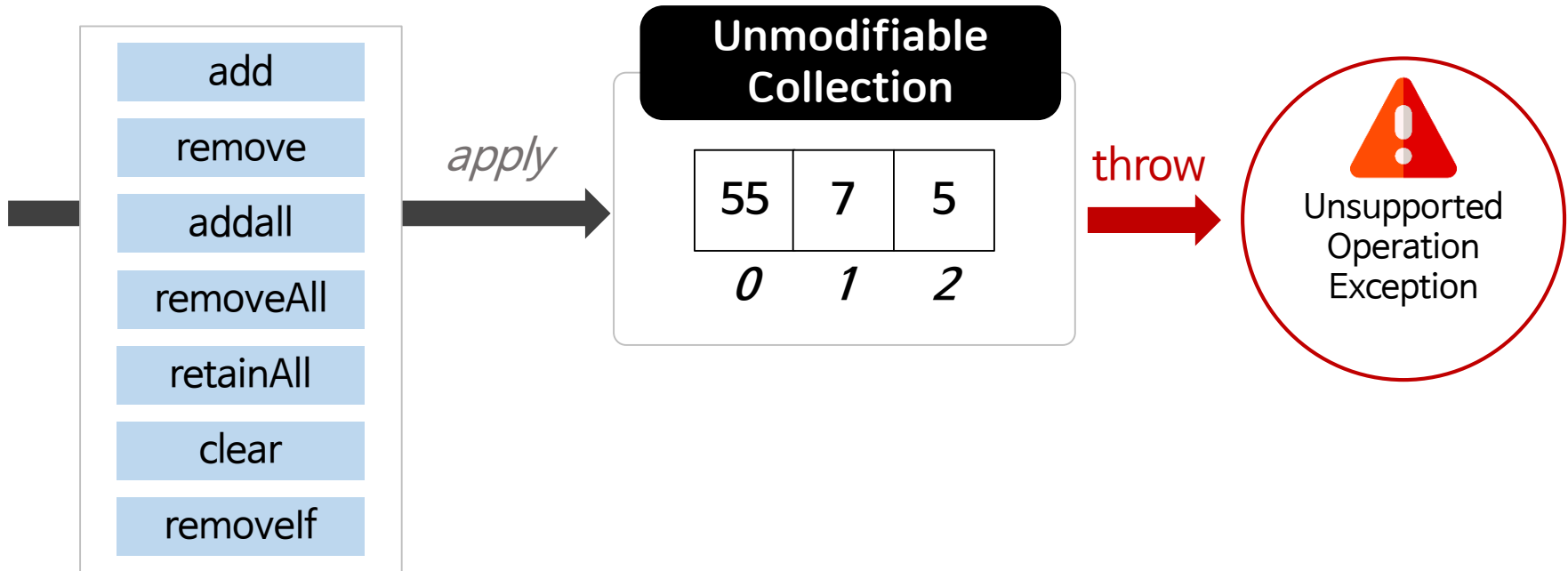
View Collections



Unmodifiable Collections



Unmodifiable Collections





Collection Interface

모든 Interface 클래스에
공통적인 것

Return Type	Method	Description
boolean	isEmpty()	Collection이 비어 있는지 확인
int	size()	Collection의 크기를 반환
boolean	add(E e)	Collection에 새로운 instance를 삽입
boolean	contains(Object o)	Collection에 o라는 instance가 있는지 확인
boolean	remove(Object o)	Collection에 o라는 instance가 있다면 삭제
Iterator<E>	iterator()	Collection을 순회할 수 있는 iterator를 반환
void	clear()	Collection을 비움
<T> T[]	toArray(T[] a)	Collection을 T타입의 배열에 담음
Stream<E>	stream()	Collection에 대한 Stream을 반환

Collection Interface

		Return Type	Method	Description
메타데이터		boolean	isEmpty()	Collection이 비어 있는지 확인
		int	size()	Collection의 크기를 반환
CREATE		boolean	add(E e)	Collection에 새로운 instance를 삽입
RETRIEVE		boolean	contains(Object o)	Collection에 o라는 instance가 있는지 확인
DELETE		boolean	remove(Object o)	Collection에 o라는 instance가 있다면 삭제
TRAVERSE		Iterator<E>	iterator()	Collection을 순회할 수 있는 iterator를 반환
UPDATE		void	clear()	Collection을 비움
		<T> T[]	toArray(T[] a)	Collection을 T타입의 배열에 담음
CRUD 연산		Stream<E>	stream()	Collection에 대한 Stream을 반환

추가적으로 전체 요소들을 순회할 수 있는 방법 제공

Collection

어떠한 개념으로 제작

CRUD, T 연산이 주는 변화



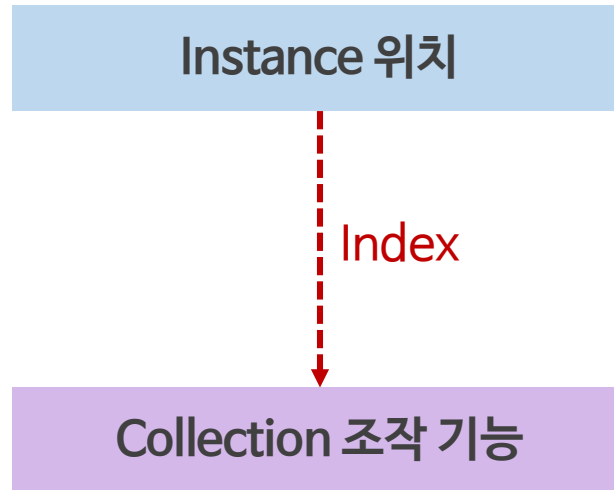
충분한 활용 가능

List

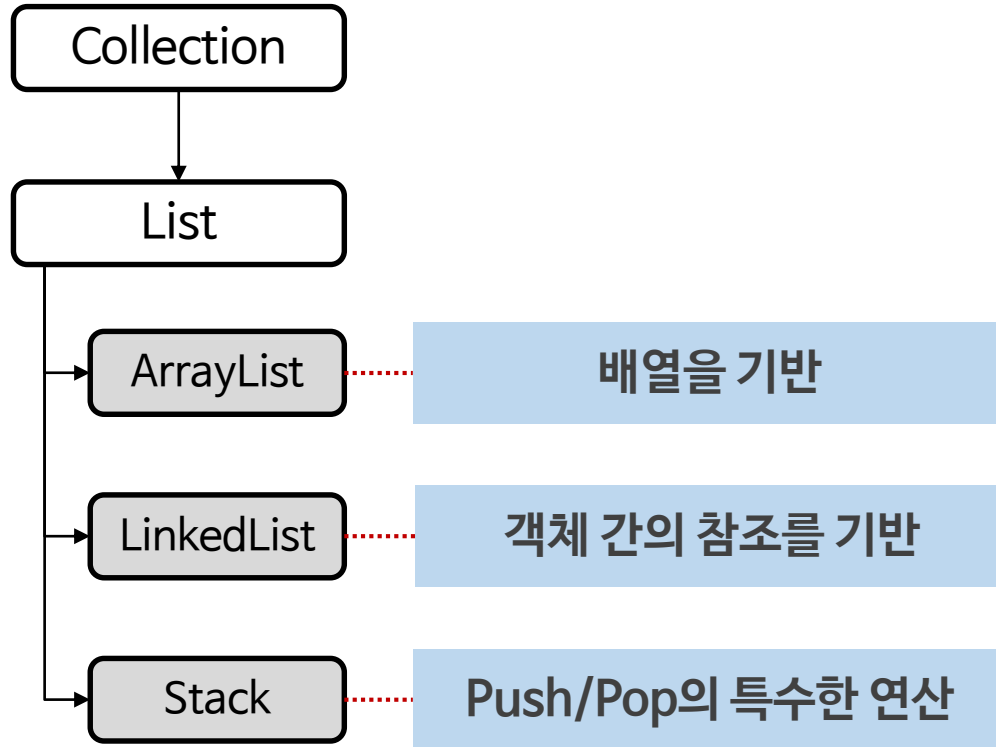
Collection 상속 Interface

순서화된 중복 허용

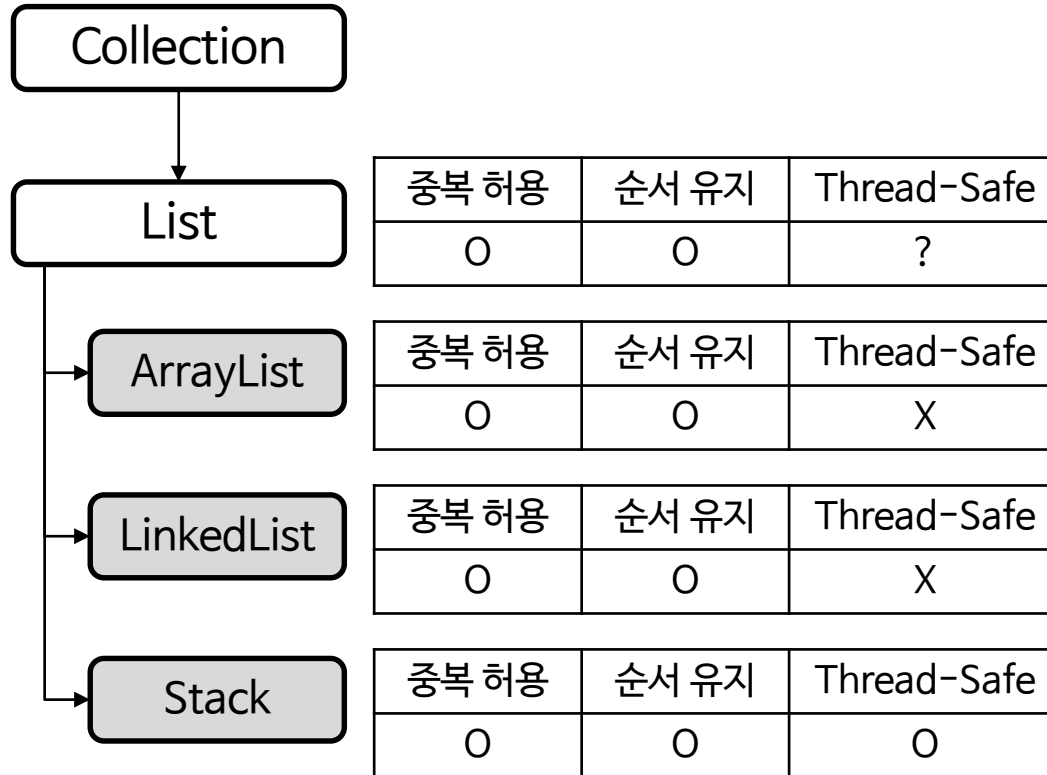
List



List



List



구현에 따라 제공하는 연산의 효율이 다를 수 있음

List의 핵심 Method

META

C

R

U

D

T

Return Type	Method	Description
boolean	isEmpty()	Collection이 비어 있는지 확인
int	size()	Collection의 크기를 반환
boolean	add(E e)	Collection에 새로운 instance를 삽입
void	add(int index, E element)	List의 특정 위치에 instance를 삽입
boolean	contains(Object o)	Collection에 o라는 instance가 있는지 확인
E	get(int index)	List의 특정 위치에 있는 instance를 추출
int	indexOf(Object o)	List에서 instance o의 위치를 찾기 (앞에서부터)
int	lastIndexOf(Object o)	List에서 instance o의 위치를 찾기 (뒤에서부터)
E	set(int index, E element)	List의 특정 위치의 instance 값을 element로 업데이트
boolean	remove(Object o)	Collection에 o라는 instance가 있다면 삭제
E	remove(int index)	List의 특정 위치에 있는 instance를 삭제
Iterator<E>	iterator()	Collection을 순회할 수 있는 iterator를 반환
ListIterator<E>	listIterator(int index)	List를 순회할 수 있는 listIterator를 반환
void	clear()	Collection을 비움
Object[]	toArray()	Collection을 배열에 담음
void	sort(Comparator<? super E> comparator)	List를 특정 비교방법에 의해 정렬함
List<E>	subList(int from, int to)	List의 부분을 가져온다
Stream<E>	stream()	Collection에 대한 Stream을 반환

Collection에서 상속된 Method 외에 List에서 추가로 정의하고 있는 Method



ArrayList

Array를 기반으로 List Interface를 구현한 것

ArrayList

Capacity (용적)이 변경 가능한 Array



ArrayList

S	E	J	O	N	G	-	M	O	O
0	1	2	3	4	5	6	7	8	9

기본값: 10

Add C



S	E	J	O	N	G	-	M	O	O	C			
<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>...</i>	

Trim to size 11



S	E	J	O	N	G	-	M	O	O	C
0	1	2	3	4	5	6	7	8	9	10

ArrayList

S	E	J	O	N	G	-	M	O	O
0	1	2	3	4	5	6	7	8	9

Add C



S	E	J	O	N	G	-	M	O	O	C			
0	1	2	3	4	5	6	7	8	9	10	11	...	

특정 Threshold 초과 시
자동 증가

수동 증가 가능
: ensureCapacity

Trim to size 11



S	E	J	O	N	G	-	M	O	O	C
0	1	2	3	4	5	6	7	8	9	10

ArrayList

S	E	J	O	N	G	-	M	O	O
0	1	2	3	4	5	6	7	8	9

Add C



S	E	J	O	N	G	-	M	O	O	C			
0	1	2	3	4	5	6	7	8	9	10	11	...	

Trim to size 11



S	E	J	O	N	G	-	M	O	O	C
0	1	2	3	4	5	6	7	8	9	10

ArrayList의 실제 크기로
Capacity 줄일 수 있음
: trimToSize

Remind

Java Collection Framework 개념

ArrayList 개념

자료 출처

#01 flaticon, 2021, URL : https://www.flaticon.com/free-icon/alarm_1672451