



ArrayList CRUD 연산(3)

학습내용

- 1 ArrayList CRUD 연산 소개(3)
- 2 ArrayList CRUD 연산 실습(3)

학습목표

- ArrayList CRUD 연산의 개념을 설명할 수 있다.
- ArrayList CRUD 연산을 활용할 수 있다.

ArrayList

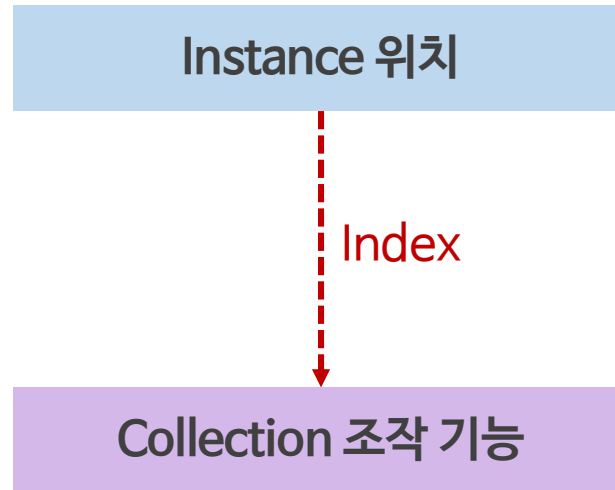


순서화된 중복을
허용하는 Collection



배열을 기반으로
List 구현

ArrayList



Collection



List



ArrayList



LinkedList



Stack

중복 허용	순서 유지	Thread-Safe
0	0	?

중복 허용	순서 유지	Thread-Safe
0	0	X

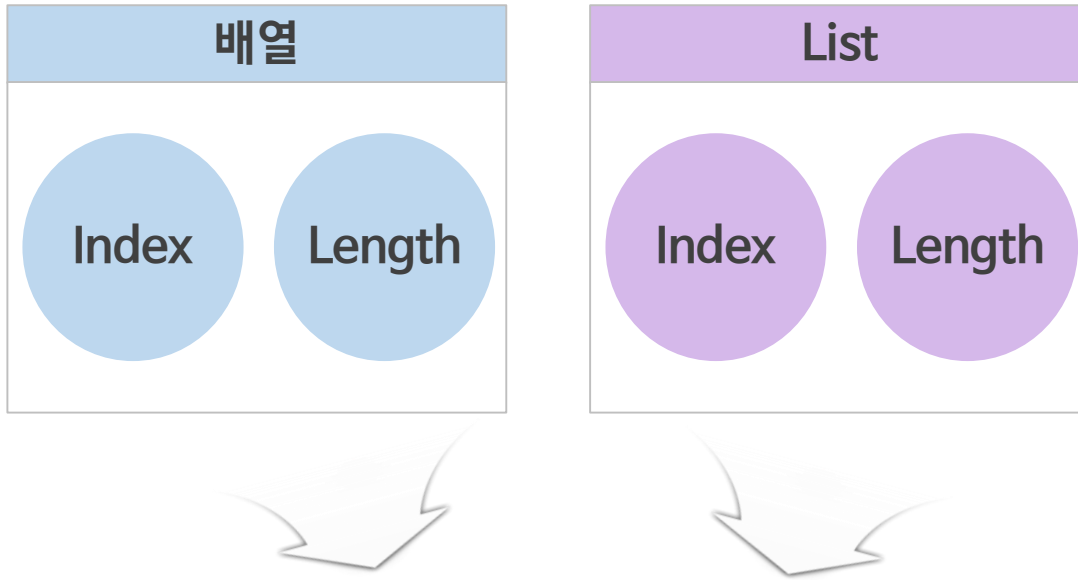
중복 허용	순서 유지	Thread-Safe
0	0	X

중복 허용	순서 유지	Thread-Safe
0	0	0

구현에 따라 제공하는 연산의 효율이 다름

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 T타입의 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



일반적 For Loop를 통해
모든 요소 소비 가능

Collection 구현 중에
Index가 없기도 함

Return Type	Method	Description
Iterator<E>	iterator()	Collection을 순회할 수 있는 iterator를 반환

Iterator

Return Type	Method	Description
Iterator<E>	iterator()	Collection을 순회할 수 있는 iterator를 반환

값	K	M	O	O	C
Index	0	1	2	3	4

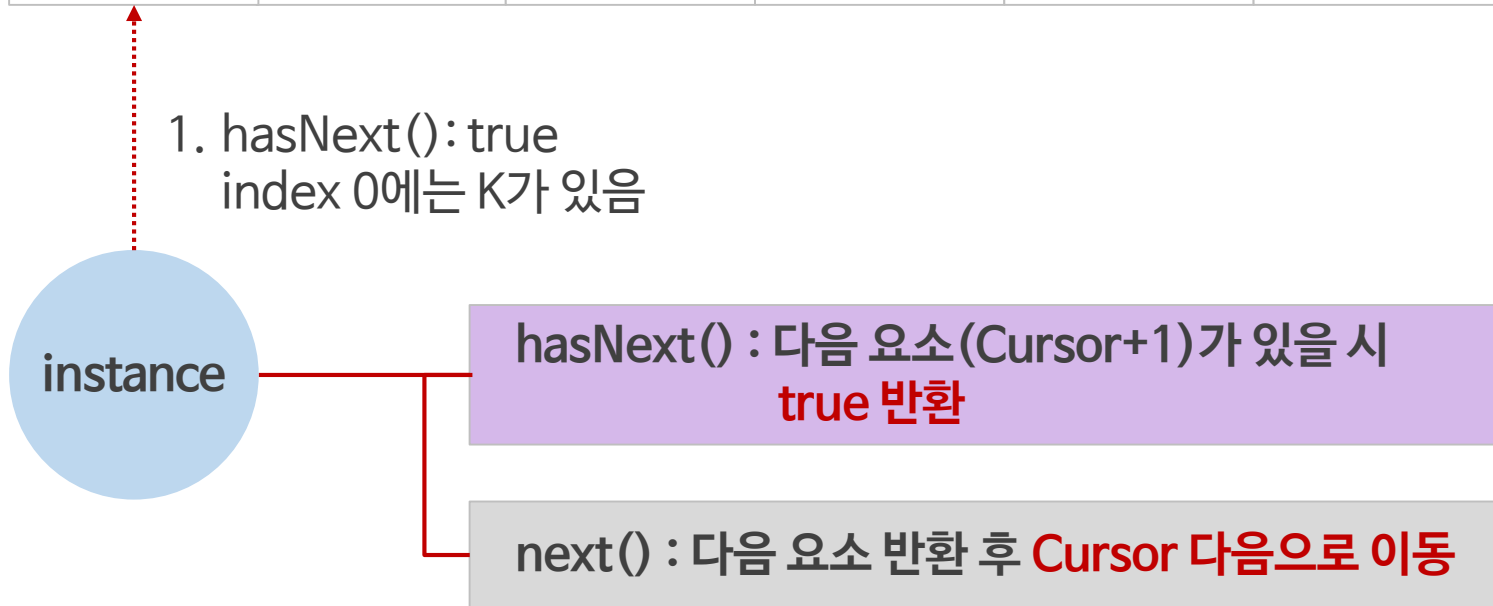
Cursor의 초기위치
:-1



Iterator

Return Type	Method	Description
Iterator<E>	iterator()	Collection을 순회할 수 있는 iterator를 반환

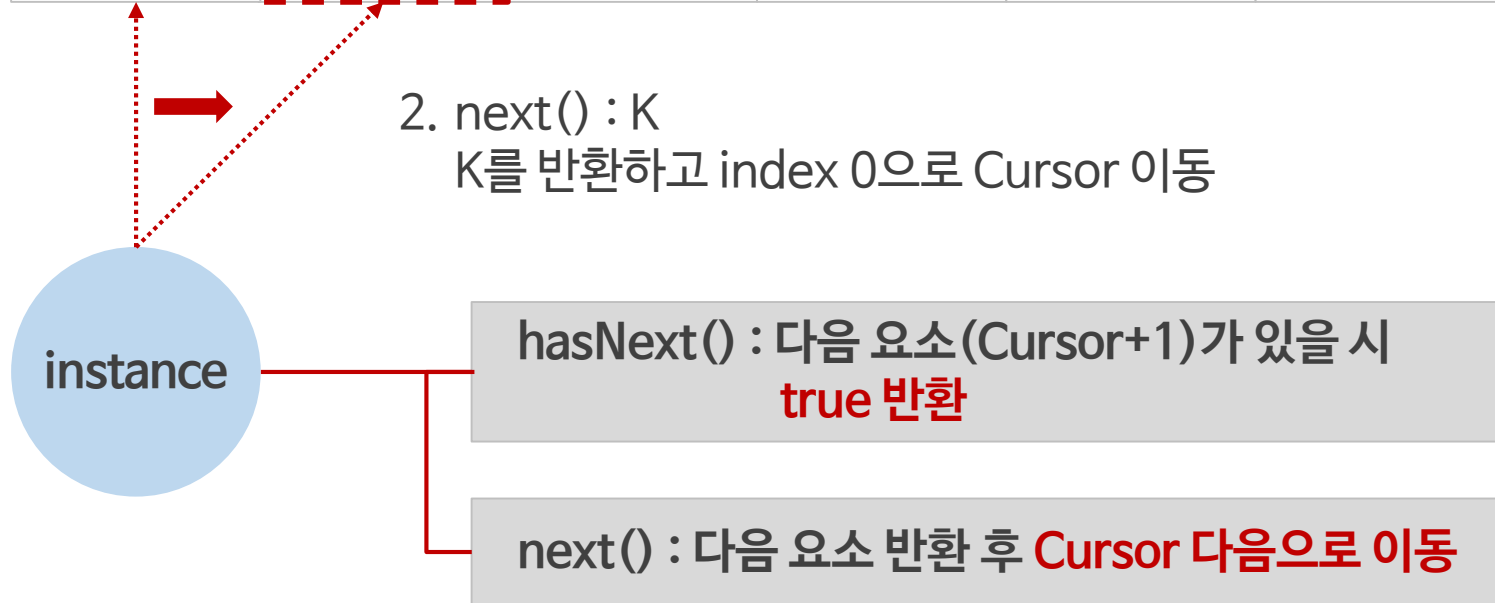
값	K	M	O	O	C
Index	0	1	2	3	4



Iterator

Return Type	Method	Description
Iterator<E>	iterator()	Collection을 순회할 수 있는 iterator를 반환

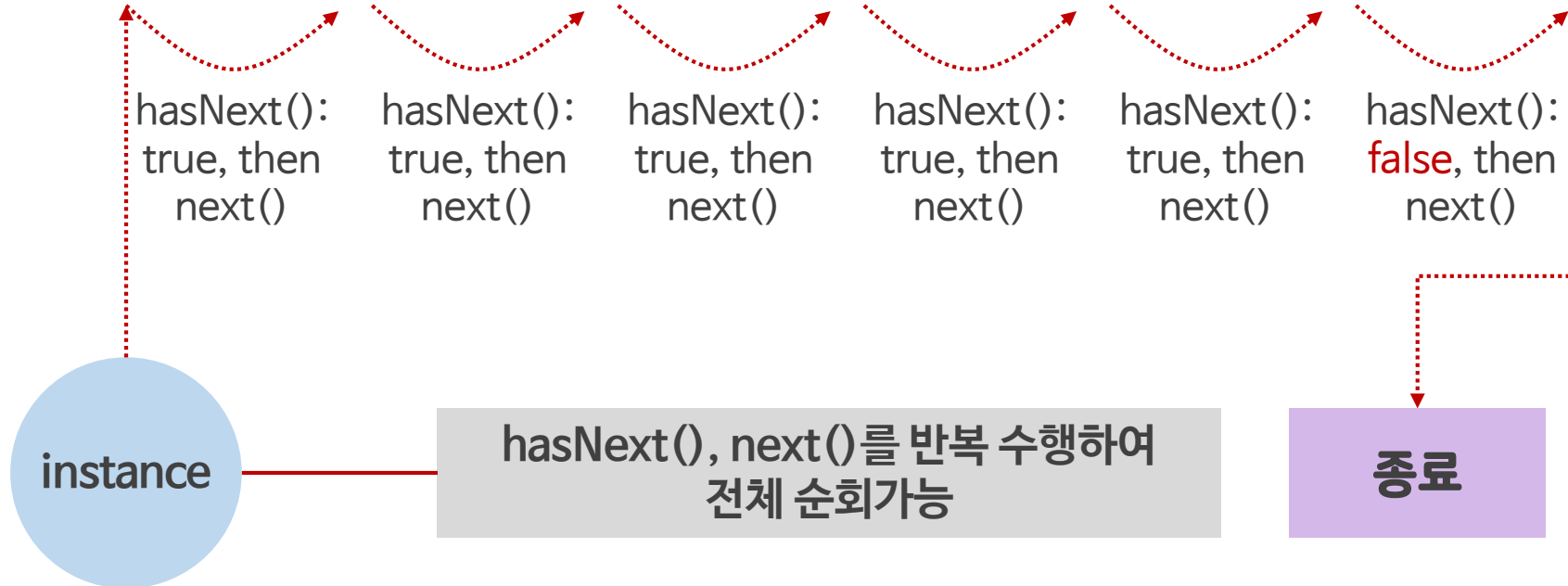
값	K	M	O	O	C
Index	0	1	2	3	4



Iterator

Return Type	Method	Description
Iterator<E>	iterator()	Collection을 순회할 수 있는 iterator를 반환

값	K	M	O	O	C
Index	0	1	2	3	4



Iterator

Return Type	Method	Description
Iterator<E>	iterator()	Collection을 순회할 수 있는 iterator를 반환

Listiterator

Iterator 보다 향상된 기능성을 갖는
Listiterator 제공

Return Type	Method	Description
ListIterator<E>	listIterator (int index)	List를 순회할 수 있는 listIterator를 반환

listIterator

Return Type	Method	Description
ListIterator<E>	listIterator(int index)	List를 순회할 수 있는 listIterator를 반환

값	K	M	O	O	C
Index	0	1	2	3	4

Cursor의 초기위치
: 4 (5-1)

생성자 ListIterator(int index)를 통해
초기 위치 설정가능

Iterator()는 ListIterator(0)과 동일

hasPrevious()

previous()

hasNext()

next()

ListIterator(5)
List의 size()

listIterator

Return Type	Method	Description
ListIterator<E>	listIterator(int index)	List를 순회할 수 있는 listIterator를 반환

값	K	M	O	O	C
Index	0	1	2	3	4

hasPrevious() : true
index 4에는 C가 있음

List의 사이즈로 초기화

ListIterator(5)
List의 size()

listIterator

Return Type	Method	Description
ListIterator<E>	listIterator(int index)	List를 순회할 수 있는 listIterator를 반환

값	K	M	O	O	C
Index	0	1	2	3	4

previous(): C
C를 반환하고 index 3으로 Cursor 이동

ListIterator(5)
List의 size()

listIterator

Return Type	Method	Description
ListIterator<E>	listIterator(int index)	List를 순회할 수 있는 listIterator를 반환

값	K	M	O	O	C
Index	0	1	2	3	4

hasPrevious(): false, then previous()
 hasPrevious(): true, then previous()
 hasPrevious(): true, then previous()
 hasPrevious(): true, then previous()
 hasPrevious(): true, then previous()
 hasPrevious(): true, then previous()

hasPrevious(), previous()를 반복 수행하여
전체 순회가능

ListIterator(5)
List의 size()

Remind

ArrayList CRUD 연산

Return Type	Method	Description
Iterator<E>	iterator()	Collection을 순회할 수 있는 iterator를 반환
ListIterator<E>	listIterator(int index)	List를 순회할 수 있는 listIterator를 반환