



ArrayList CRUD 연산(2)

학습내용

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

학습목표

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

ArrayList



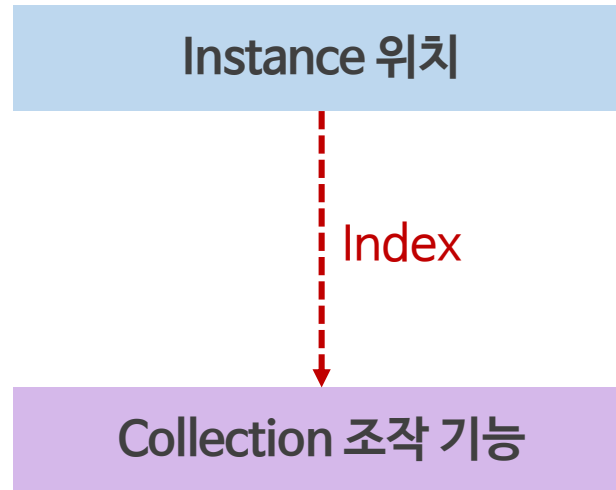
List

순서화된 중복을
허용하는 Collection

ArrayList

배열을 기반으로
List 구현

ArrayList



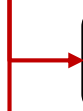
Collection



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

contains

Return Type	Method	Description
boolean	contains (Object o)	Collection에 o라는 instance가 있는지 확인

contains

myArrayList				
K	M	O	O	C
0	1	2	3	4



contains("C"):
C가 포함되었는지 확인

True

myArrayList				
K	M	O	O	C
0	1	2	3	4



contains("P"):
P가 포함되었는지 확인

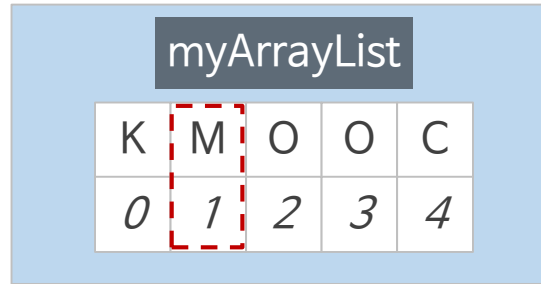
False

get(int index)

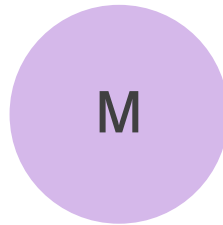
Return Type	Method	Description
E	get(int index)	List의 특정 위치에 있는 instance를 추출



get(int index)



`get(1):`
Index 1의 instance 가져오기



indexOf, lastIndexOf

List의 중복 허용

Return Type	Method	Description
int	indexOf (Object o)	List에서 instance o의 위치를 찾기 (앞에서부터)
int	lastIndexOf (Object o)	List에서 instance o의 위치를 찾기 (뒤에서부터)

indexOf, lastIndexOf

myArrayList				
K	M	O	O	C
0	1	2	3	4

indexOf("o")

: O가 최초로 발견된 Index
(뒤에서부터 찾기)

2

lastIndexOf("o")

: O가 최초로 발견된 Index
(앞에서부터 찾기)

3

set UPDATE

Return Type	Method	Description
E	set (int index, E element)	List의 특정 위치의 instance 값을 element로 UPDATE

set UPDATE

myArrayList				
K	N	O	O	C
0	1	2	3	4



set(1, "M ") :
Index 1의 요소를 M으로 갱신

myArrayList				
K	M	O	O	C
0	1	2	3	4

remove

Return Type	Method	Description
boolean	Remove (Object o)	Collection에 o라는 instance가 있다면 삭제

삭제
성공

True를 반환

ArrayList에서 재위치 작업이 일어날 수 있음

remove

myArrayList					
K	N	O	O	C	C
0	1	2	3	4	5



remove("C"):
C가 있다면 지우기
(앞에서 부터 검색, 단 한번 지움)

myArrayList				
K	M	O	O	C
0	1	2	3	4

index 기반 remove

Return Type	Method	Description
E	remove (int index)	List의 특정 위치에 있는 instance를 삭제

index 기반 remove

myArrayList					
K	N	O	O	C	C
0	1	2	3	4	5



remove(4):
Index 4 instance 삭제
(4이후의 값이 앞당겨짐)

myArrayList				
K	M	O	O	C
0	1	2	3	4

Remind

ArrayList CRUD 연산

Return Type	Method	Description
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