



List 인터페이스와 호환 가능한 배열 기반 자료구조 설계 및 활용



학습내용

- 1 List 인터페이스와 호환 가능한 배열 기반 자료구조 설계 및 구현
- 2 직접 만든 배열기반 List를 통해 실 세계 데이터 분석 수행

학습목표

- List 인터페이스와 호환가능한 배열 기반 자료 구조 설계 및 구현할 수 있다.
- 직접 만든 배열 기반 List를 통해 실 세계 데이터 분석을 수행하고 한계를 설명할 수 있다.

MyArrayList



순서화된 중복을
허용하는 Collection



배열을 기반으로
List Interface 구현

Capacity 동적 변화

ArrayList

META
C
R
U
D
T

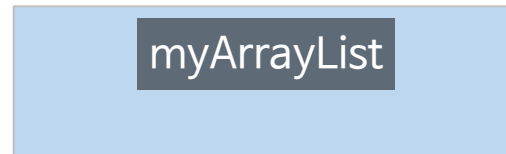
Return Type	Method	Description
boolean	isEmpty()	Collection이 비어 있는지 확인
int	size()	Collection의 크기를 반환
boolean	add(E e)	Collection에 새로운 instance를 삽입
void	add(int index, E element)	List의 특정 위치에 instance를 삽입
boolean	contains(Object o)	Collection에 o라는 instance가 있는지 확인
E	get(int index)	List의 특정 위치에 있는 instance를 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
<T> T[]	toArray(T[] a)	Collection을 T타입의 배열에 담음

clear

Return Type	Method	Description
void	clear()	Collection을 비움



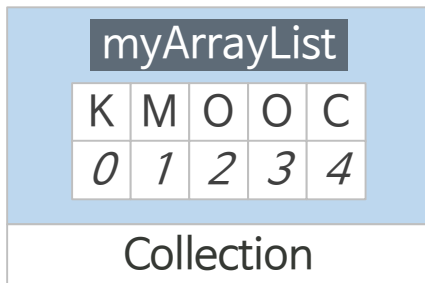
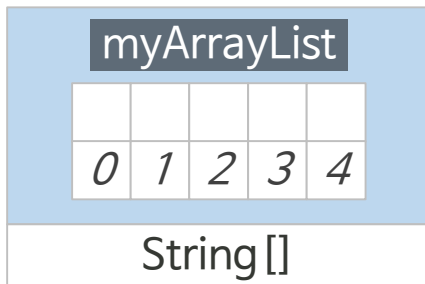
clear() :



Object [] data
= new Object [] ; 수행

toArray

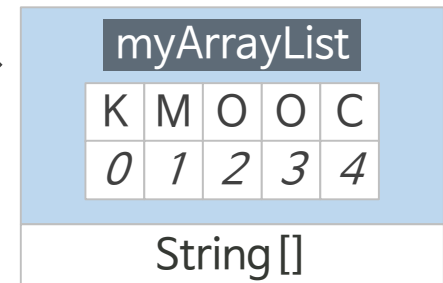
Return Type	Method	Description
$\langle T \rangle T []$	<code>toArray(T [] a)</code>	Collection을 T타입의 배열에 담음



Data 반환



toArray
myArrayList
data in
newArray





MyArrayList를 이용한 데이터 분석 수행

이벤트의 수
구하기

최소 사람 ID
구하기

최대 사람 ID
구하기

Remind

ArrayList CRUD 연산

Return Type	Method	Description
void	clear()	Collection을 비움
Object[]	toArray()	Collection을 T타입의 배열에 담음