



List 인터페이스의 참조기반 자료구조 설계(5)

- 2가지 메소드를 통한 구현 -



학습내용

- 1 List 인터페이스의 참조기반 자료구조 설계
- 2 MyLinkedList의 2가지 메소드 구현

학습목표

- List 인터페이스의 참조기반 자료구조를 설계할 수 있다.
- MyLinkedList의 2가지 메소드 구현할 수 있다.

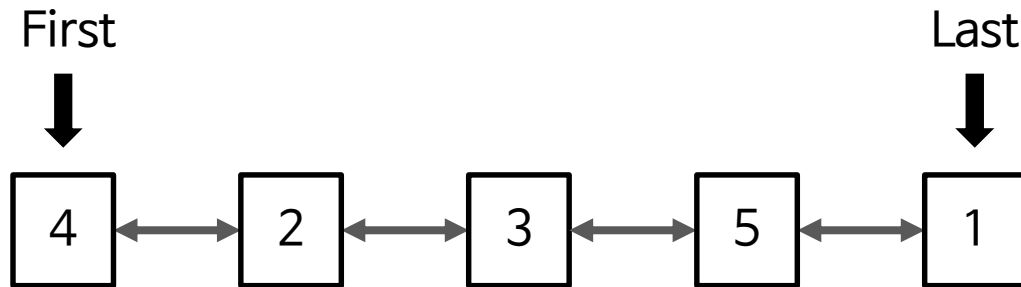
META
C
R
U
D
T

Return Type	Method	Description
boolean	isEmpty()	Collection이 비어 있는지 확인
int	size()	Collection의 크기를 반환
boolean	add(E e)	Collection에 새로운 instance를 삽입 (Capacity 초과시 Exception 발생)
void	add(int index, E element)	List의 특정 위치에 instance를 삽입
boolean	offer(E e)	Queue에 새로운 instance를 마지막에 삽입 (Capacity 초과시 false 반환)
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	element()	Queue의 처음 instance를 반환 (비어있을 때 Exception 발생)
E	peek()	Queue의 처음 instance를 반환 (비어있을 때 null 반환)
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
E	remove()	Queue의 처음 instance를 지우고 반환(비어있을 때 Exception 발생)
E	poll()	Queue의 처음 instance를 지우고 반환 (비어있을 때 null 반환)
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

toArray

Return Type	Method	Description
Object[]	toArray()	Collection을 배열에 담음

toArray



toArray()

Object[]

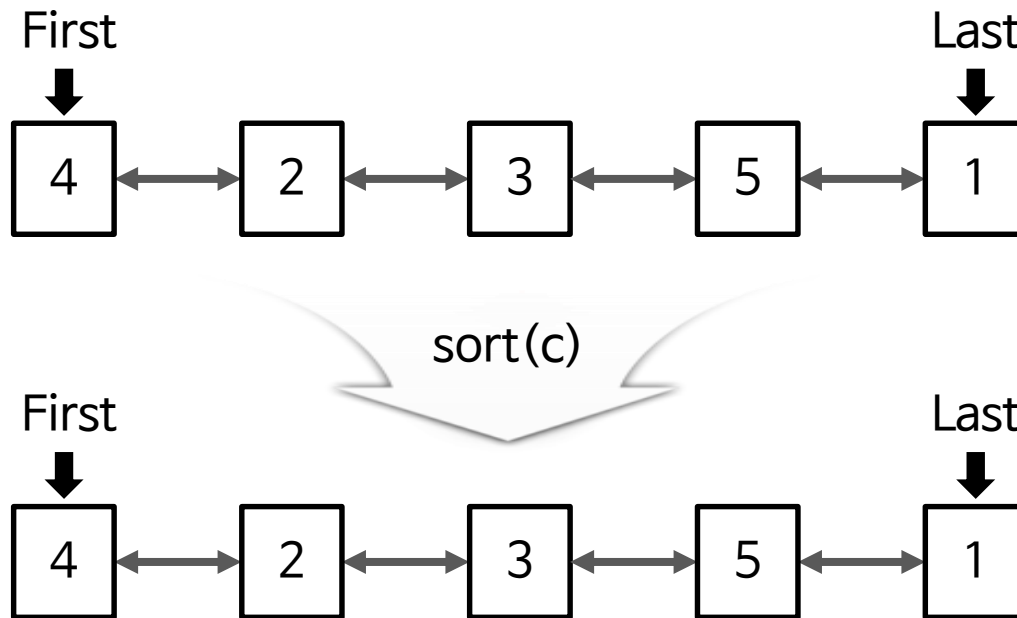
값	4	2	3	5	1
Index	0	1	2	3	4

sort

Return Type	Method	Description
void	sort(Comparator<? super E> comparator)	List를 특정 비교방법에 의해 정렬함

sort

toArray와 listIterator 구현으로 동작 보장



Remind

MyLinkedList 구현

Return Type	Method	Description
Object []	toArray()	Collection을 배열에 담음
void	sort(Comparator<? super E> comparator)	List를 특정 비교방법에 의해 정렬함