



# List 인터페이스의 참조기반 자료구조 설계(2)

- 3가지 메소드 구현 -

## 학습내용

- 1 List 인터페이스의 참조기반 자료구조 설계
- 2 MyLinkedList의 3가지 메소드 구현

## 학습목표

- List 인터페이스의 참조기반 자료구조를 설계할 수 있다.
- MyLinkedList의 3가지 메소드를 구현할 수 있다.

# Linked

참조 기반

List  
Interface

Queue  
Interface

**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

Return Type	Method	Description
boolean	add(E e)	Collection에 새로운 instance를 삽입
String	toString()	MyLinkedList의 문자열 표현을 반환한다.
생성자	MyLinkedList()	빈 LinkedList를 생성
생성자	MyLinkedList(Collection c)	LinkedList를 Collection c의 요소를 가져와 생성

## add(E e)

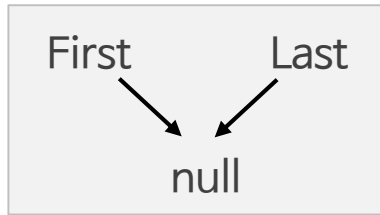
Return Type	Method	Description
boolean	add(E e)	Collection에 새로운 instance를 삽입

MyLinkedList가  
비어 있을 때

Vs.

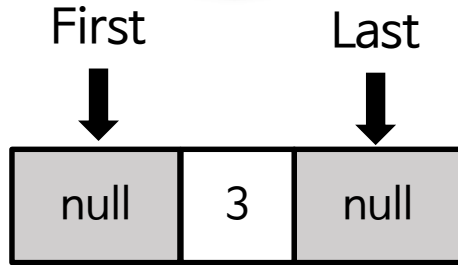
MyLinkedList가  
비어 있지 않을 때

# MyLinkedList가 비어 있을 때



size = 0

add(3)



size = 1

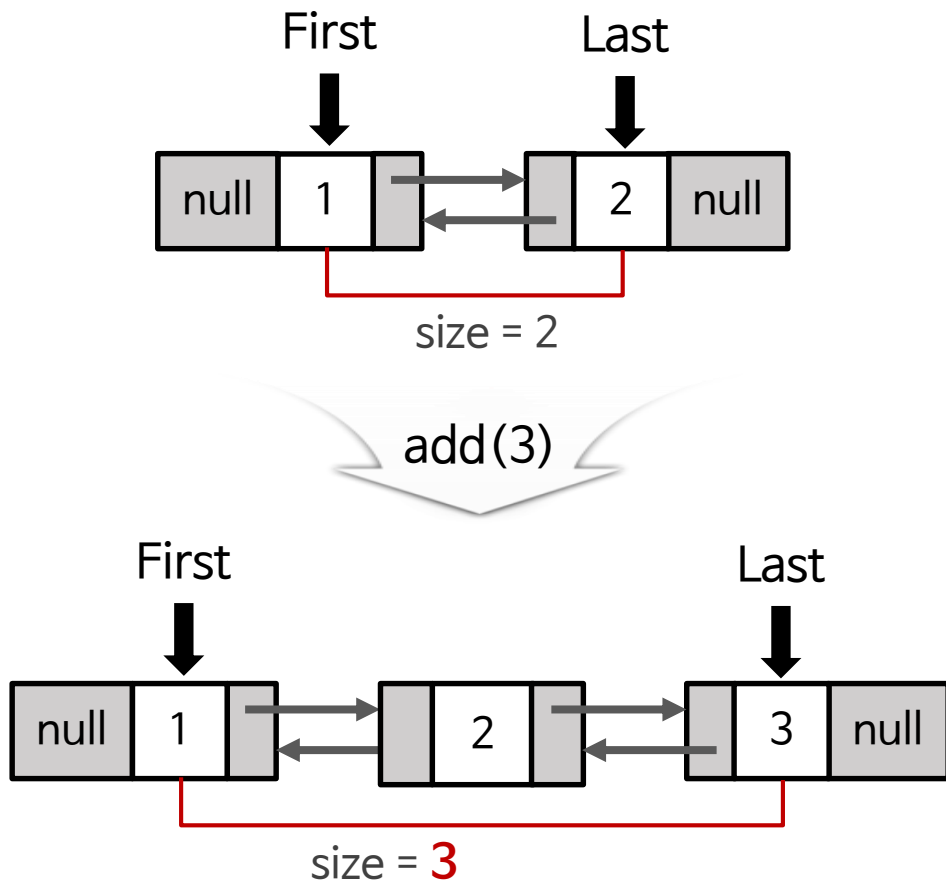
1. MyNode 생성

2. First, Last 갱신

3. size++



# MyLinkedList가 비어있지 않을 때



## 1. MyNode newNode 생성

- prev = last;
- item = e;
- next = null;

## 2. last.next 참조 갱신

- last.next = newNode;

## 3. last 갱신

## 4. size++

## toString()

Return Type	Method	Description
String	toString()	MyLinkedList의 문자열 표현을 반환한다.



Size = 0

[]를 반환

Size  $\neq$  0

first부터 시작하여  
getNext()가 null일  
때까지 순회하며 문자  
열을 생성하여 반환

## MyLinkedList 생성

Return Type	Method	Description
생성자	LinkedList()	빈 LinkedList를 생성
생성자	LinkedList (Collection c)	LinkedList를 Collection c의 요소를 가져와 생성

# for-each loop

```
LinkedList<Integer> linkedList = new LinkedList<Integer>();
```

2. 주소할당

JVM Heap

1. 빈 List 생성

LinkedList  
[ ]

```
Collection<Integer> collection = arrayList; // [0,1,2,3,4] ArrayList  
LinkedList<Integer> linkedList = new LinkedList<Integer>(collection);
```

2. 주소할당

1. Collection으로 List 생성

LinkedList  
[0,1,2,3,4]

# Remind

---

add(E e)

toString

생성자