

Data Engineering

# 데이터엔지니어링

[리스트 기반 데이터엔지니어링 (3)]



# LinkedList 소개



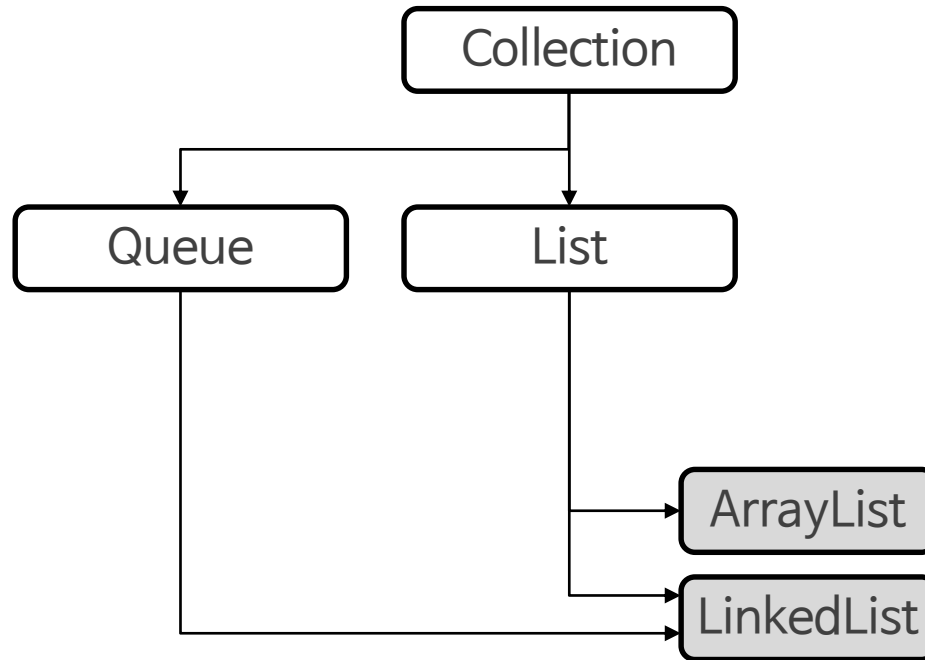
## 학습내용

- 1 LinkedList 구성 인터페이스
- 2 LinkedList 연산의 소개와 실습

## 학습목표

- LinkedList를 구성하는 인터페이스 관계를 이해할 수 있다.
- LinkedList에서 제공하는 연산을 이해하고 활용할 수 있다.

# java.util.LinkedList 계층구조



# ArrayList



List

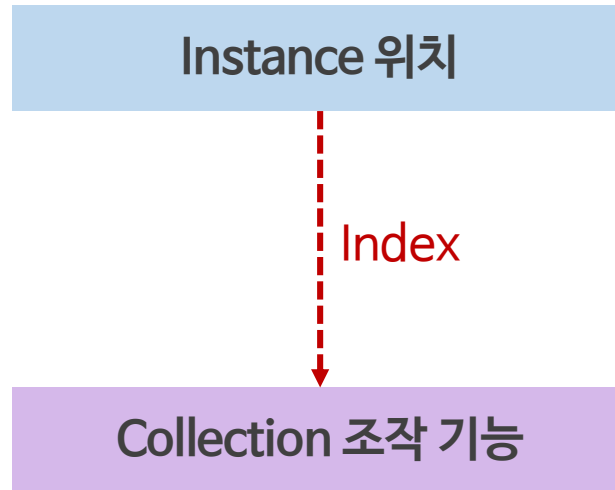
순서화된 중복을  
허용하는 Collection

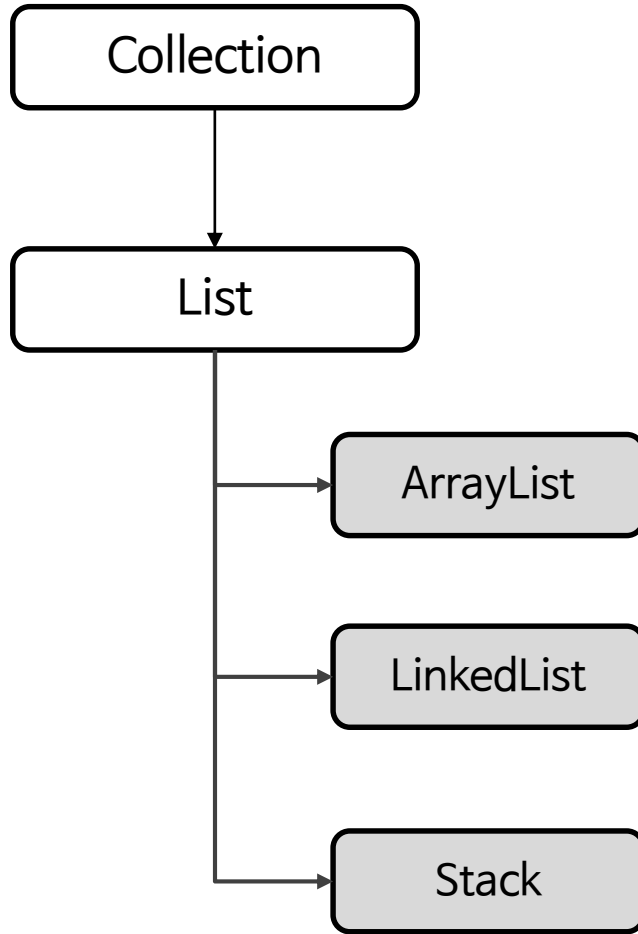


ArrayList

배열을 기반으로  
List 구현

# ArrayList





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**

# LinkedList

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 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





# LinkedList

ArrayList와 동일하게 동작

---

Method 각각의 효율이 다를 수 있음



LinkedList

||

Queue

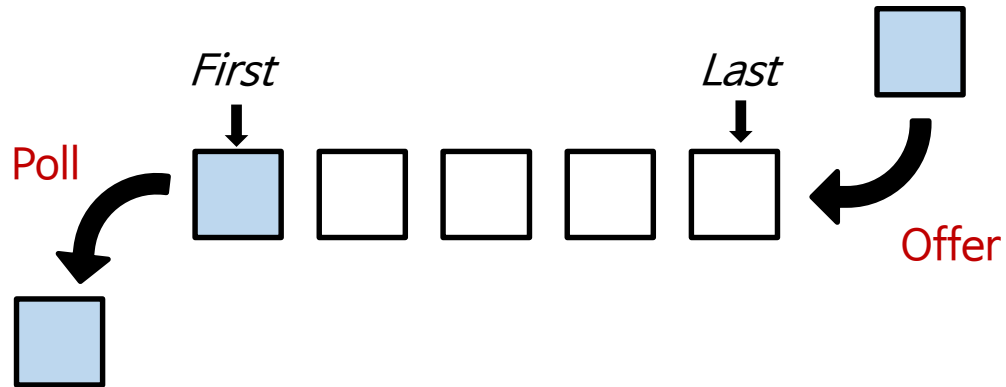
# Queue

삽입

추출

한 방향

# Queue



**First In First Out (FIFO)**

# Queue



버퍼

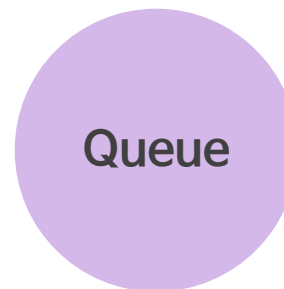
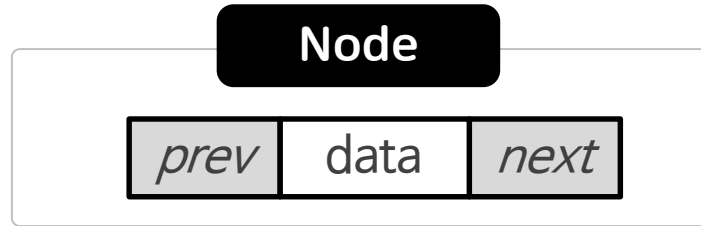


스케줄러

# Queue

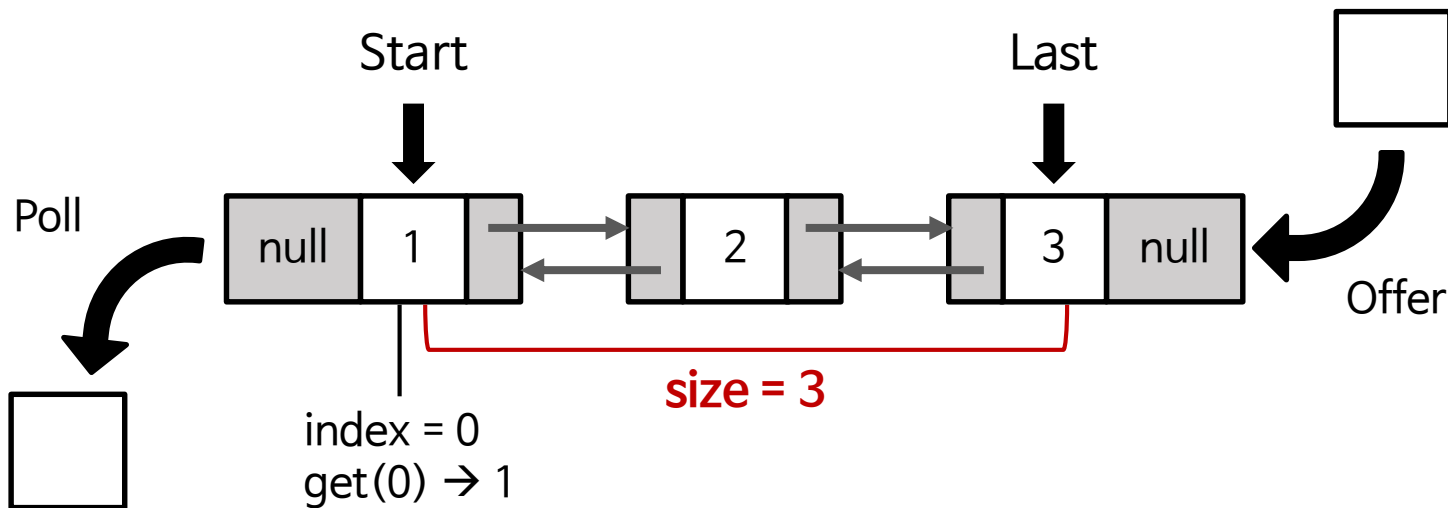
		Return Type	Method	Description
META  C  R  D  T	[	boolean	isEmpty()	Collection이 비어 있는지 확인
		int	size()	Collection의 크기를 반환
	[	boolean	add(E e)	Collection에 새로운 instance를 삽입 (Capacity 초과시 Exception 발생)
		boolean	offer(E e)	Queue에 새로운 instance를 마지막에 삽입 (Capacity 초과시 false 반환)
	[	boolean	contains(Object o)	Collection에 o라는 instance가 있는지 확인
		E	element()	Queue의 처음 instance를 반환 (비어있을 때 Exception 발생)
		E	peek()	Queue의 처음 instance를 반환 (비어있을 때 null 반환)
	[	boolean	remove(Object o)	Collection에 o라는 instance가 있다면 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
		void	clear()	Collection을 비움
		Object[]	toArray()	Collection을 배열에 담음
		Stream<E>	stream()	Collection에 대한 Stream을 반환

# LinkedList



# 참조 기반 리스트 구현

참조 개수에 따른 다양한 방식



**Doubly LinkedList**



# LinkedList



List

각 Node는  
Index로서 접근 가능



Queue

뒤쪽으로 값을 삽입하고  
앞에서 값을 추출

# Remind

---

LinkedList 구성 인터페이스

LinkedList의 개념과 연산