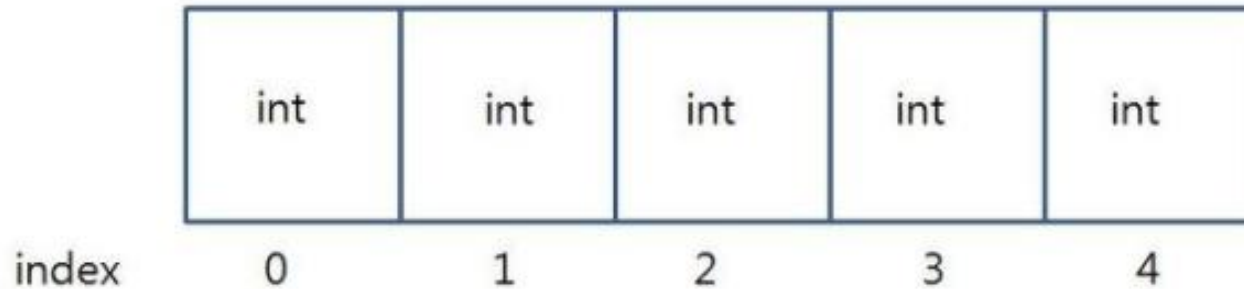


리스트

김수진

배열

배열 : `int[] array = new int[5];`



- 내부 인덱스를 가지고 데이터 접근 -> 인덱스 번호가 데이터의 순서
- 검색 시 빠르게 접근 가능
- 0부터 시작하기 때문에 첫 데이터에 쉽게 접근, 삽입 시 순차적으로 인덱스 값을 증가시키므로 길이를 통해 마지막 데이터도 쉽게 접근 -> 스택, 큐 구현시 유용
- 참조를 위한 추가적인 메모리 할당이 필요 없다. (Linked list의 경우 node라는 객체를 생성해야 함)
- 중복 허용, but 하면 안됨.

배열의 삽입

```
int[] arr = new int[20];  
int size = 0;  
  
for(int i=0; i<arr.length; i++){  
    arr[i] = i+1;  
    size++;  
}
```

- 삽입할 인덱스를 가진 배열의 객체에 삽입할 자료를 대입

배열의 검색

```
int searchKey = 5;  
for(int i=0; i<size; i++){  
    if(arr[i] == searchKey){  
        System.out.println(i + "번째에 " + searchKey + " 을 찾았다.");  
        break;  
    }  
}
```

- 찾고자 하는 값과 비교하여 검색

배열의 삭제

```
int deleteKey = 5;
for(int i=0; i<size; i++){
    if(arr[i] == deleteKey){
        for(int k=i; k<size-1; k++){
            arr[k] = arr[k+1];
        }
        size--;
        break;
    }
}
```

- 삭제하고자 하는 값과 같으면 앞으로 한 칸씩 밀어주고 배열의 크기를 하나 감소시킨다.

배열의 한계

- 생성 시 지정한 크기 변환 불가능
- 삽입과 삭제 시 앞으로 당겨주거나 뒤로 밀어주는 과정을 작업해야 함
-> 데이터 수 / 2 만큼 이동이 발생하게 됨.
- 삽입, 삭제가 어려워 고정된 크기를 가지는 일만 수행하는게 바람직함.
- 길이를 넘는 인덱스를 참조하려고 할 때 오류가 발생함.

사용하기 위해서는 고정된 크기를 할당해 놓아야 하지만, 프로그래밍 할 때 크기를 항상 알 수는 없기 때문에 List가 탄생하게 됨.

리스트

- 데이터를 순차적으로 나열해 놓은 집합 (추상적인 개념임)
- 비슷한 성질의 데이터를 저장하여 그룹화하기 위해 사용함.
- 순서 고려
- 빈 엘리먼트를 허용하지 않기 때문에 자료 형의 개수가 가변하는 상황에서 사용 가능
- 중복 허용
- 구현 방법 : 배열 vs 리스트 (ArrayList vs LinkedList)

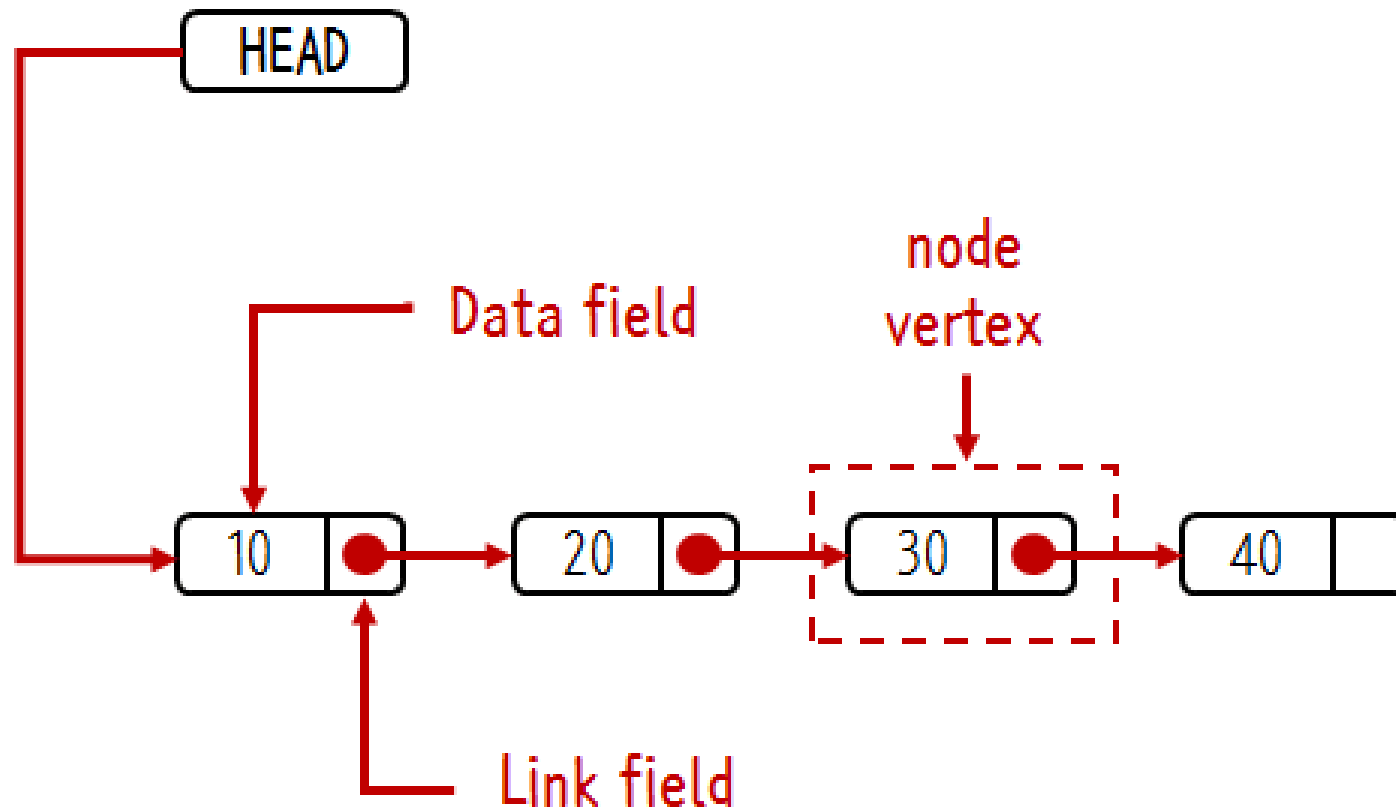
리스트 자료형

- ArrayList
- LinkedList

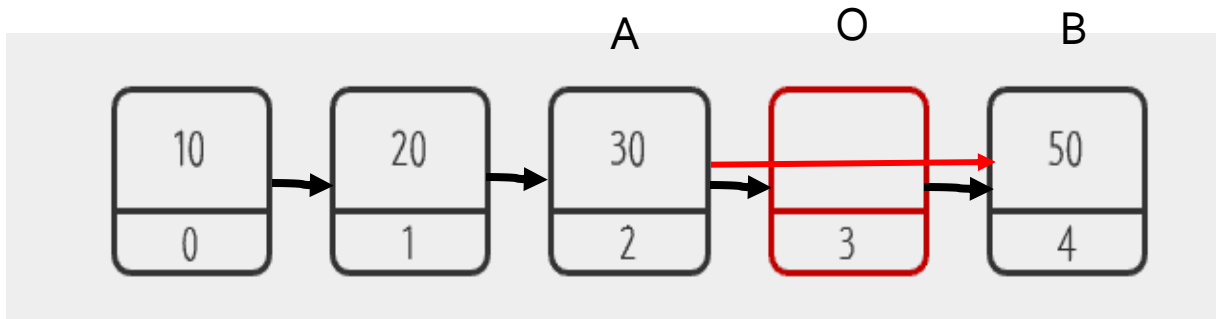
	추가/삭제	인덱스 조회
Array List		
Linked List		

인덱스를 이용해서 데이터를 가져오는 것이 빈번하다면 ArrayList, 데이터의 추가/삭제가 빈번하다면 LinkedList가 효과적

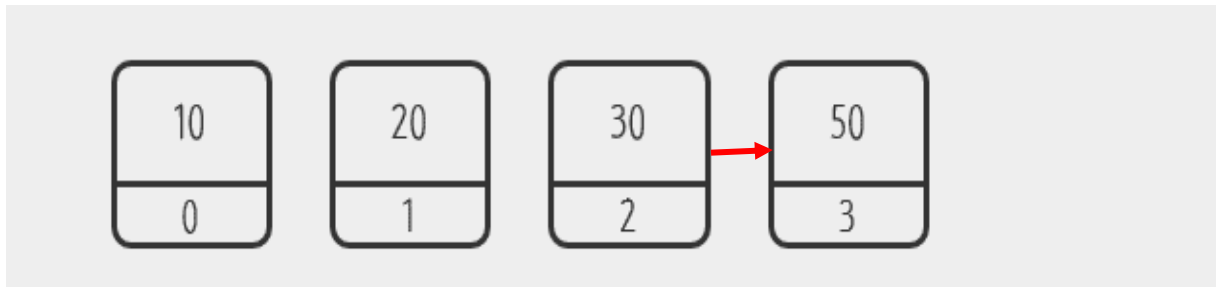
LinkedList



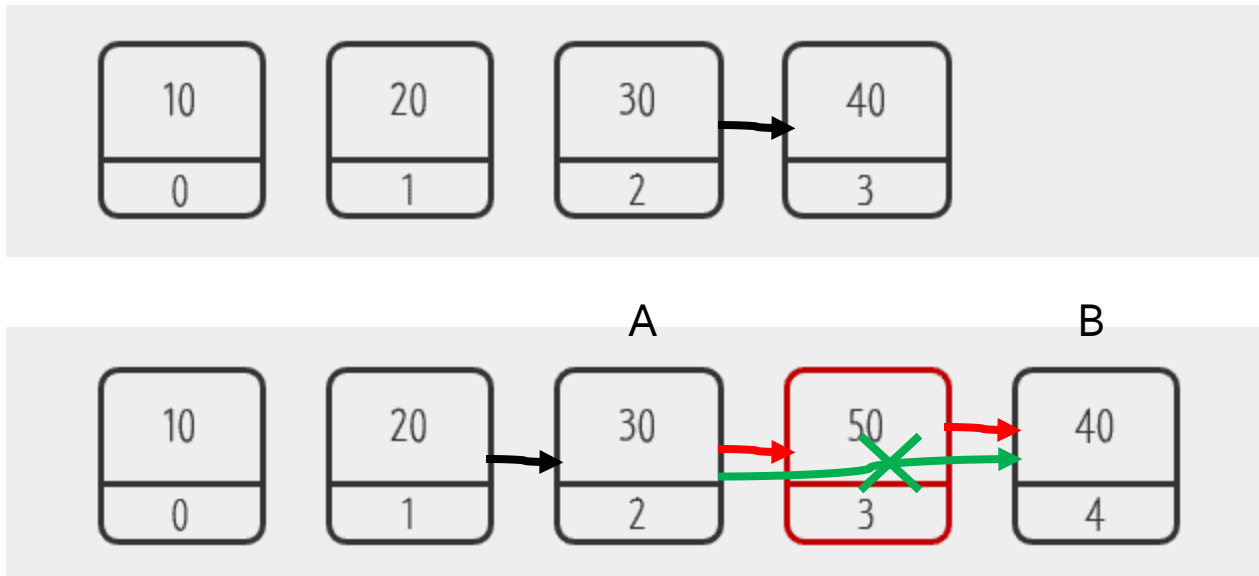
LinkedList 삭제



A.next = A.next.next;
or
A.next = O.next;



LinkedList 삽입



```
Node temp = new Node(50)  
temp.next = A.next;  
A.next = temp;
```

Built in List

- **add**

ex) `employee.add("수진"); employee.add(0, "수진");`

- **get**

ex) `employee.get(0);`

- **contains**

ex) `boolean isContain = employee.contains("수진");`

- **remove**

ex) `employee.remove("수진");`
`employee.remove(0);`

- **size**

ex) `int size = employee.size();`