Review

순환

순환 (recursion)
- 정의하려는 개념 자체를 정의 속에 포함하여 이용
- 종류
  - 직접 순환 : 함수가 직접 자신을 호출
  - 간접 순환 : 다른 제 3의 함수를 호출하고 그 함수가 다시 자신을 호출
- 순환 방식의 적용
  - 분할 정복(divide and conquer)의 특성을 가진 문제에 적합
  - 어떤 복잡한 문제를 직접 간단하게 풀 수 있는 작은 문제로 분할하여 해결하려는 방법.
  - 분할한 문제는 원래의 문제와 그 성질이 같기 때문에 푸는 방법도 동일
- 순환 함수의 명령문 골격
  - if (simplest case) then solve directly
    else {make a recursive call to a simpler case};

순환

Factorial (n!)

정의

n=0 : 1

$n \geq 1$: $n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1 = n \cdot (n-1)!$

순환 함수의 표현

```
factorial(n)
        // n은 음이 아닌 정수
         if  (n ≤ 1) then return 1
         else return (n*factorial(n-1));
end factorial()
```

순환

피보나치 수열 (Fibonacci sequence)

각 항은 바로 직전 두 항의 합으로 만들어짐

순환 정의

$n=0 : f_0 = 0$
$n=1 : f_1 = 1$
$n \geq 2 : f_n = f_{n-1} + f_{n-2}$

순환 함수의 표현

```
fib(n)
        if (n ≤ 0) then return 0;
        else if (n=1) then return 1
        else return (fib(n –1)  + fib(n –2));
 end fib()
```

배열

선형 리스트 (linear list)

순서를 가진 원소들의 순열(sequence)

물리적 순서가 아닌 원소의 특성에 의한 논리적 순서를 의미

리스트는 기본적으로 순서 개념을 가지므로 선형 리스트라고 볼 수 있음

리스트 $L=(e_1, e_2, \ldots, e_n)$

L은 리스트 이름, e_i는 리스트 원소

공백 리스트(empty list, 원소가 하나도 없는 리스트)의 표현 : L=( )

리스트의 각 원소는 선행자(predecessor)와 후속자(successor)를 가짐

예

자료 구조 강의 요일 = (월요일, 수요일, 금요일)

토요일 강의 과목 = ()

배열

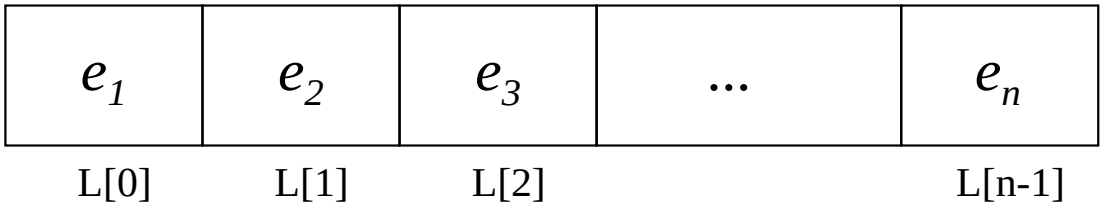
배열을 사용해 표현 (순차 표현 리스트)

리스트 원소 e_i와 e_{i+1}이 인덱스 i-1과 i에 대응되게 연속적으로 저장

원소의 물리적 순서로 논리적 순서를 나타냄 (순서를 표시하기 위한 특별한 장치가 필요 없음)

삽입, 삭제 시에 후속 원소들을 한자리씩 밀거나 당겨야 하는 오버헤드가 치명적인 약점

$$L = (e_1, e_2, \ldots, e_n)$$



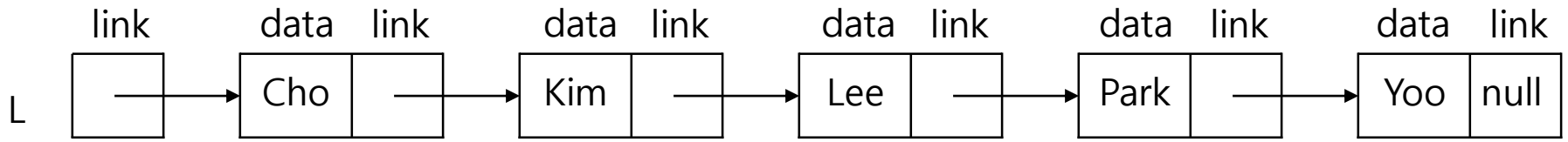
배열

List.java

ListTest.java

연결 리스트

- 연결 리스트
  - 링크를 이용해 표현한 리스트

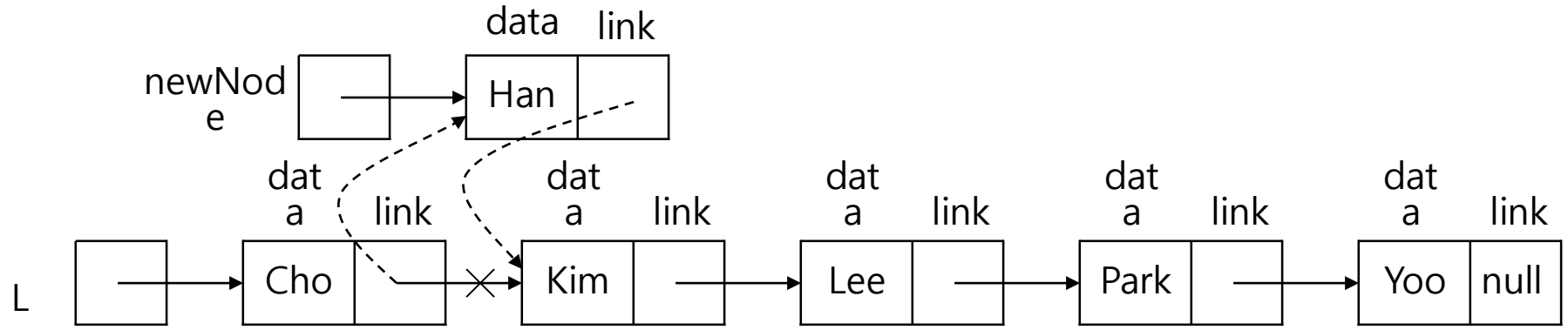


- 스트링 리스트에 대한 연결 표현을 위한 노드 구조

```
class ListNode {
        String name;
        ListNode link;
}
```

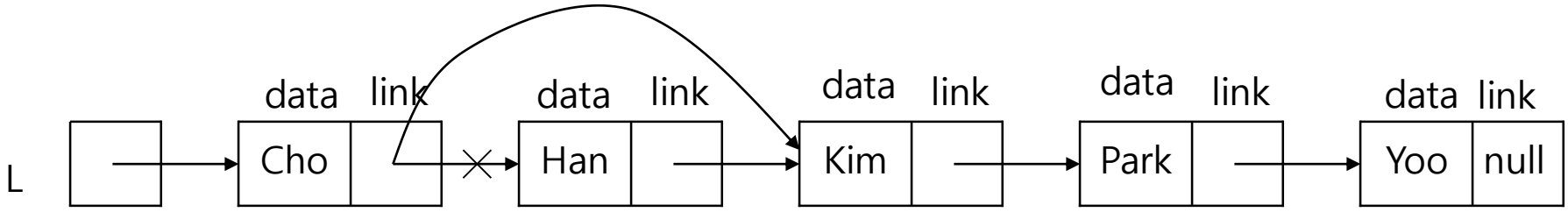
연결 리스트

- **원소 삽입 알고리즘**
  - 예) 리스트 L에 원소 “Han”을 “Cho”와 “Kim” 사이에 삽입
    - 1. 공백노드를 획득함. newNode라는 변수로 가리키게 함
    - 2. newNode의 data 필드에 “Han”을 저장
    - 3. “Cho”를 저장하고 있는 노드의 link 값을 newNode의 link 필드에 저장
      (즉 “Kim”을 저장하고 있는 노드의 주소를 newNode의 link에 저장)
    - 4. “Cho”를 저장한 노드의 link에 newNode의 포인터 값을 저장
  - 리스트의 기존 원소들을 이동시킬 필요 없음
  - 부수적인 link 필드를 위해 저장 공간을 추가로 사용



연결 리스트

- **원소 삭제 알고리즘**
  - 예) 리스트 L에서 원소 "Han"을 삭제
    - 1. 원소 "Han"이 들어 있는 노드의 선행자 찾음 ("Cho"가 들어있는 노드)
    - 2. 이 선행자의 link에 "Han"이 들어있는 노드의 link 값을 저장



연결 리스트


ListNode.java


LinkedList.java


LinkedListTest.java