6주차 1차시 원형 연결 리스트

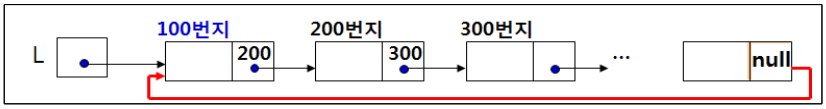
【학습목표】

1. 원형 연결 리스트의 특징에 대하여 설명할 수 있다.
2. 원형 연결 리스트의 삽입과 삭제 연산에 대하여 설명할 수 있다.

학습내용1 : 원형 연결 리스트

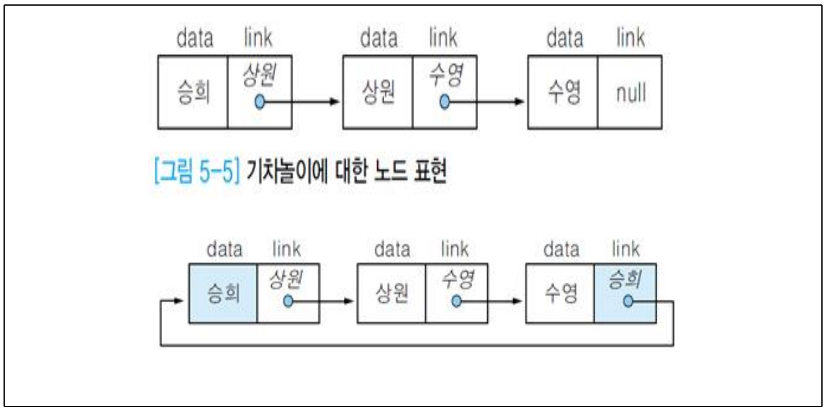
1. 정의

- 단순 연결 리스트에서 마지막 노드가 리스트의 첫 번째 노드를 가리키게 하여 리스트의 구조를 원형으로 만든 연결 리스트
- 단순 연결 리스트의 마지막 노드의 링크 필드에 첫 번째 노드의 주소를 저장하여 구성
- 링크를 따라 계속 순회하면 이전 노드에 접근 가능



2. 선형 vs. 원형 기차놀이

[그림 5-4] 기차놀이 : 뽑은 이름표대로 기차 연결하기 [그림 5-51] 원형 기차놀이



[그림 5-5] 기차놀이에 대한 노드 표현

학습내용2 : 원형 연결 리스트의 삽입

1. 원형 연결 리스트의 삽입 연산

- 마지막 노드의 링크를 첫 번째 노드로 연결하는 것만 제외하고는 단순 연결 리스트의 삽입 연산과 동일

2. 첫 번째 노드로 삽입하기

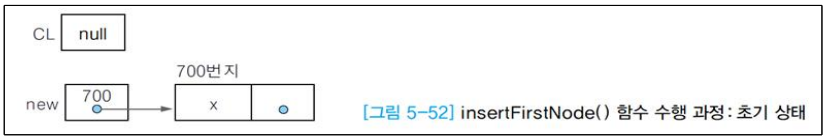
- 원형 연결 리스트 CL에 x값을 갖는 노드 new를 삽입하는 알고리즘

알고리즘 5-8 원형 연결 리스트의 첫 번째 노드 삽입 알고리즘

```
insertFirstNode(CL, x)
    new ← getNode();
    new.data ← x;
    if (CL = null) then {                  // ❶ 공백 리스트인 경우
        CL ← new;                          // ❶-ⓐ
        new.link ← new;                    // ❶-ⓑ
    }
    else{                                  // ❷ 공백 리스트가 아닌 경우
        temp ← CL;                         // ❷-ⓐ
        while (temp.link ≠ CL) do          // ❷-ⓑ
            temp ← temp.link;
        new.link ← temp.link;              // ❷-ⓒ
        temp.link ← new;                   // ❷-ⓓ
        CL ← new;                          // ❷-ⓔ
    }
end insertFirstNode()
```

❶ [원형 리스트가 공백 리스트인 경우]

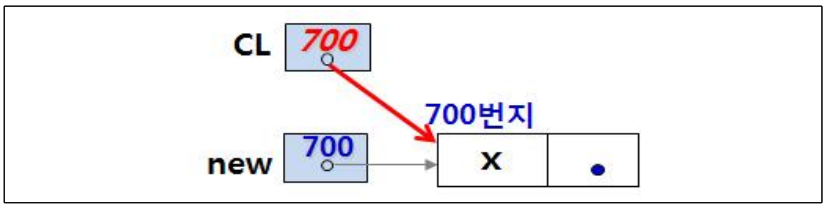
- 삽입하는 노드 new가 리스트의 첫 번째 노드이자 마지막 노드



[그림 5-52] insertFirstNode() 함수 수행 과정 : 초기 상태

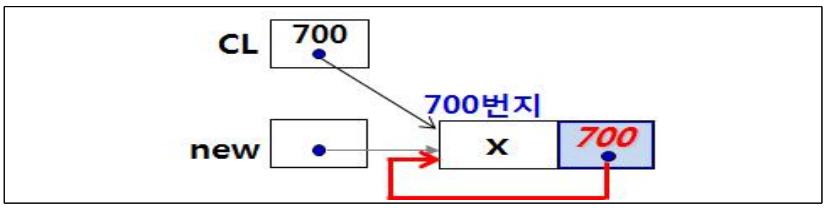
❶-ⓐ CL ← new;

- 포인터 CL이 노드 new를 가리키도록 한다



❶-ⓑ new.link ← new;

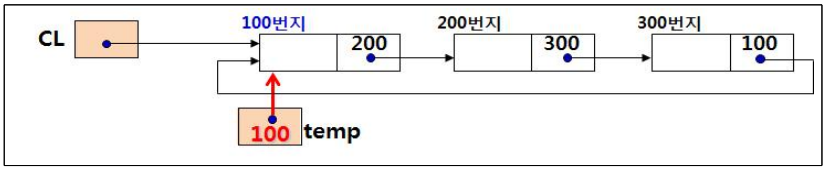
- 새 노드의 주소를 링크 필드에 저장하여 노드 new의 링크가 자기 자신을 가리키도록함으로서 원형 연결 리스트 CL의 첫 번째 노드이자 마지막 노드가 되도록 지정한다



❷ [원형 리스트가 공백 리스트가 아닌 경우]

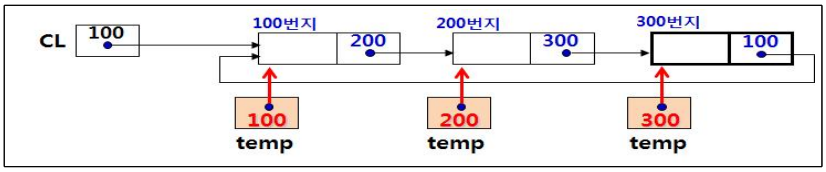
❷-ⓐ temp ← CL;

- 리스트가 공백이 아닌 경우 첫 번째 노드의 주소를 임시 순회 포인터 temp에 저장하여 노드 순회의 시작점을 지정한다



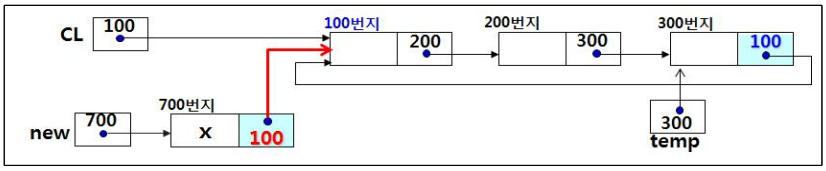
❷-ⓑ **while** (temp.link ≠ CL) **do** temp ← temp.link;

- while 반복문을 수행하여 순회 포인터 temp 링크를 따라 **마지막 노드까지 이동**



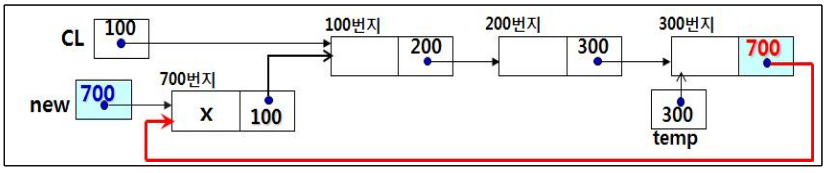
❷-ⓒ new.link ← temp.link;

- 리스트의 **마지막 노드의 링크 값을 노드 new의 링크에 저장**하여 **노드 new가 노드 temp의 다음 노드를 가리키게** 한다. 리스트 CL이 원형 연결 리스트이므로 마지막 노드의 다음 노드는 리스트의 첫 번째 노드가 된다



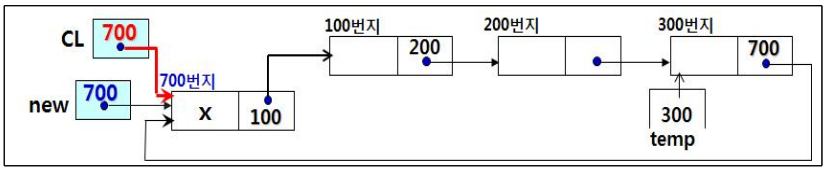
❷-ⓓ temp.link ← new;

- 노드 new의 값을 포인터 temp가 가리키고 있는 마지막 노드의 링크에 저장하여 리스트이 마지막 노드가 노드 new로 연결되도록 한다

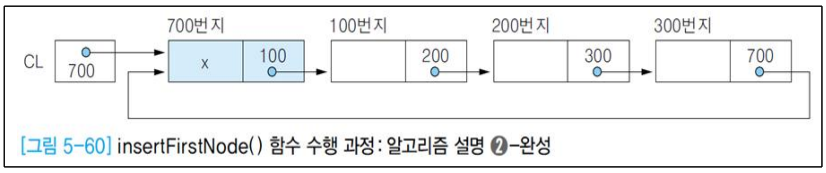


❷-ⓔ CL ← new;

- 포인터 new의 값을 리스트 시작 포인터 CL에 설정하여 노드 new가 리스트의 첫 번째 노드가 되도록 지정한다



⇒ 알고리즘 수행결과 **삽입한 노드 new는** 리스트 포인터 CL과 첫 번째 노드 사이에 삽입되면서 리스트의 **새로운 첫 번째 노드**가 되었고, **리스트의 마지막 노드와 연결하여 원형 연결 리스트 상태를 유지**한다



[그림 5-60] insertFirstNode() 함수 수행 과정 : 알고리즘 설명 ❷-완성

3. 중간 노드로 삽입하기

* 원형 연결 리스트 CL에 x값을 가진 노드 new를 포인터 pre가 가리키는 노드의 다음 노드로 삽입하는 알고리즘

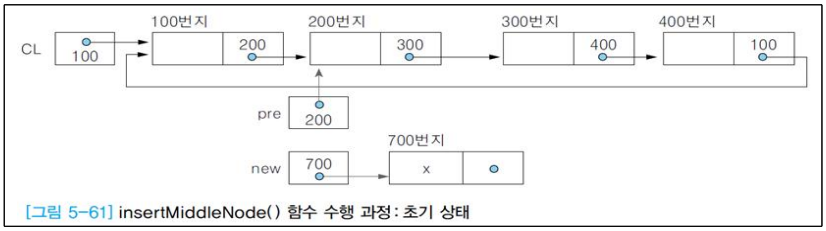
알고리즘 5-9 원형 연결 리스트의 중간 노드 삽입 알고리즘

```
insertMiddleNode(CL, pre, x)
        new ← getNode();
        new.data ← x;
        if (CL=null) then {                  // ❶ 공백 리스트인 경우
              CL ← new;
              new.link ← new;
        }
        else {                               // ❷ 공백 리스트가 아닌 경우
              new.link ← pre.link;           // ❷-ⓐ
              pre.link ← new;                // ❷-ⓑ
        }
end insertMiddleNode()
```

❶ [원형 리스트가 공백 리스트인 경우]

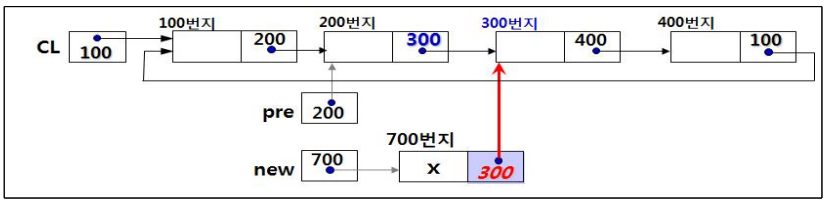
- [알고리즘 5-8]에서 공백 리스트에 첫 번째 노드를 삽입하는 경우와 동일하다

❷ [원형 리스트가 공백 리스트가 아닌 경우]

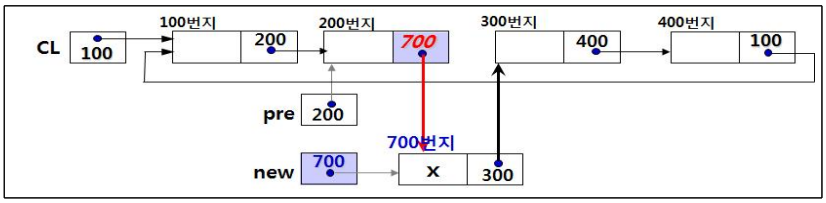


[그림 5-61] insertMiddleNode() 함수 수행 과정 : 초기 상태

❷-ⓐ new.link ← pre.link;
- 노드 pre의 다음 노드로 new를 삽입하기 위해서 먼저 노드 pre의 다음 노드(pre.link)를 노드 new의 링크 필드(new.link)에 연결한다



❷-ⓑ pre.link ← new;
- 노드 new의 값(삽입할 노드 주소)을 노드 pre의 링크 필드에 저장하여 노드 pre가 노드 new를 가리키도록 한다



학습내용3 : 원형 연결 리스트의 삭제

1. 원형 연결 리스트의 삭제 연산

- 원형 연결 리스트 CL에서 포인터 pre가 가리키는 노드의 다음 노드를 삭제하고 삭제한 노드는 자유 공간 리스트로 반환한다

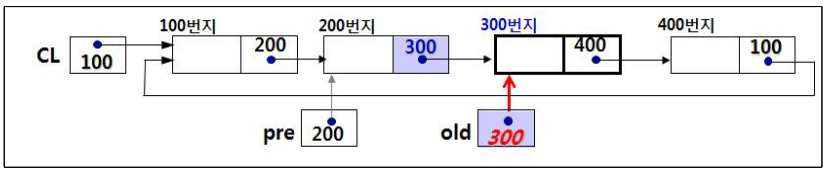
- 포인터 old는 삭제할 노드를 지시한다

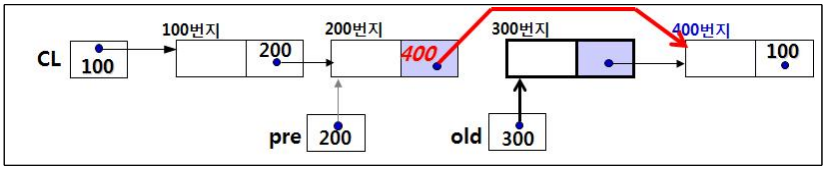
알고리즘 5-10 원형 연결 리스트의 노드 삭제 알고리즘

```
deleteNode(CL, pre)
    if (CL = null) then error;
    else {
        old ← pre.link;                 // ❶
        pre.link ← old.link;            // ❷
        if (old = CL) then              // ❸
            CL ← old.link;              // ❸-ⓐ
        returnNode(old);                // ❹
    }
end deleteNode()
```

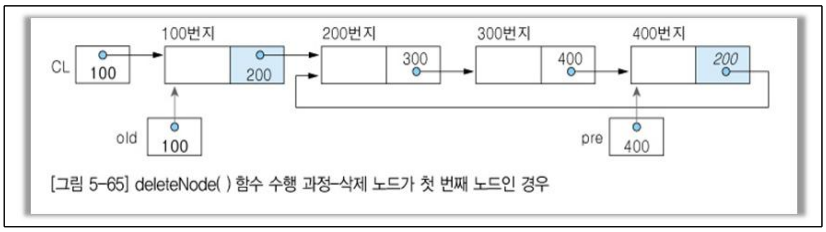
❶ old ← pre.link;
- 노드 pre의 다음 노드(pre.link)를 삭제할 노드 old로 지정한다

❷ pre.link ← old.link;
- 삭제할 노드 old의 다음 노드(old.link)를 노드 pre의 다음 노드(pre.link)로 지정한다

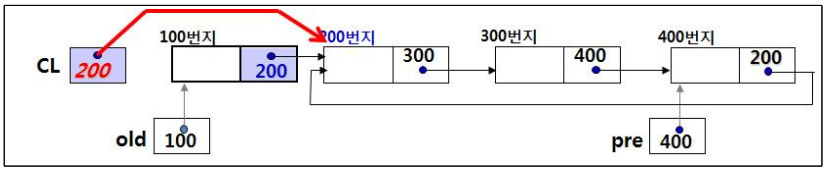




❸ [삭제할 노드 old가 원형 연결 리스트의 첫 번째 노드인 경우]



❸-ⓐ CL ← old.link;
- 첫 번째 노드를 삭제할 경우 노드 old의 링크 값을 리스트 포인터 CL에 저장하여 두 번째 노드가 리스트의 첫 번째 노드가 되도록 조정 필요



❹ returnNode(old);
- 삭제한 노드 old를 자유 공간 리스트에 반환한다

【학습정리】

1. 단순 연결 리스트에서 마지막 노드가 리스트이 첫 번째 노드를 가리키게 하여 리스트 구조를 원형으로 만든 연결 리스트를 원형 연결 리스트라고 한다.

2. 단순 연결 리스트의 마지막 노드의 링크 필드에 첫 번째 노드의 주소를 설정함으로써 원형 연결 리스트를 구성한다.