자료구조 실습04

Data Structures Lab04

Lab04 예제(1/2)

◎ 목표: Sorted Linked List 설계 및 구현

◎ 내용:

- ☞ NodeType를 이용한 Generic Sorted Linked List class 정의 및 구현
- ☞ Template을 이용한 Generic data type과 포인터을 이용한 Sorted Linked List Application 구현

◎ 방법

- ☞ 주어진 solution 코드를 분석하여 Sorted Linked List Application 구현
- ☞ 구현한 List를 template을 이용한 Generic class로 변환

예제: SortedLinkedList ADT(1/2)

```
template <typename T>
class LinkedList
{
public:
      LinkedList();                  // Constructor
     ~LinkedList();                  // Destructor

      void MakeEmpty();              // List를 비움..
      int GetLength() const;         // 리스트가 보유하고 있는 item 개수 반환
      int Add(T item);               // 새로운 레코드를 리스트에 삽입.
      int Get(T &item);              // Primary key를 기준으로 데이터를 검색하고 해당 데이터를 가져옴
      void ResetList();              // 레코드 포인터 초기화
      void GetNextItem(T &item); // Current Pointer 가 다음 node 를 가리키도록 이동 후 해당 레코드를 가져옴.

 private:
      NodeType<T> *m_pList;          // 리스트 포인터
      NodeType<T> *m_pCurPointer;    // current pointer
      int m_nLength;                 // 리스트에 저장된 레코드 수
};
```

예제: SortedLinkedList ADT(2/2)

◎ Declaration use “ struct ‘NodeType’ ”

```
template <typename T>
struct NodeType
{
    T info;                 // 노드에서 관리할 레코드
    NodeType *next;         // 다음 노드를 가리키는 포인터
};
```

예제: console

◎ List를 테스트할 driver는 다음과 같이 작성함

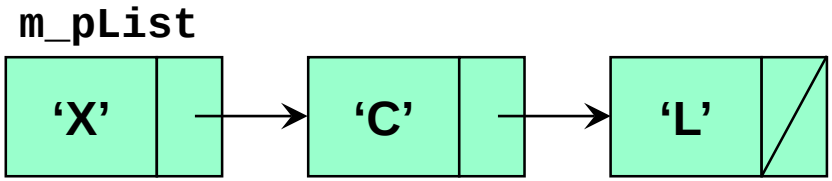
```
--- ID – Command -----
        1 : Insert Item
        2 : Delete Item
        3 : Replace Item
        4 : Retrieve Item
        5 : Display all Item
        0 : Quit

Choose a Command -->
```

Lab04: Reference (1/2)

AddItem ‘B’

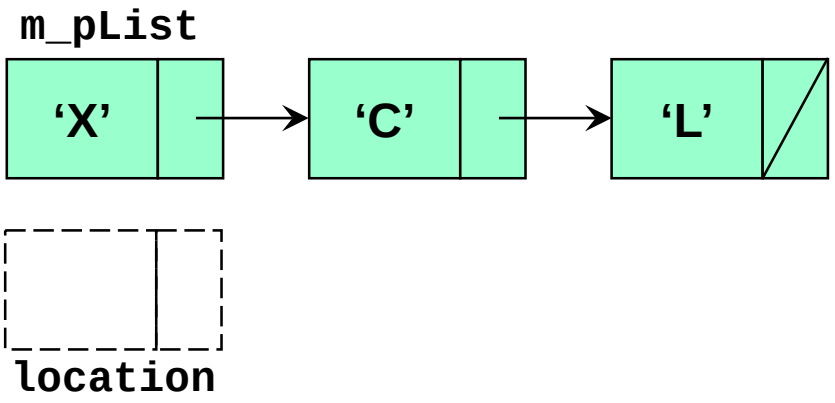
```
newItem = ‘B’;
NodeType<T> *location;
location = new  NodeType<T>;
location->info = newItem;
location->next = m_pList;
m_pList = location
```



Lab04: Reference (1/2)

AddItem ‘B’

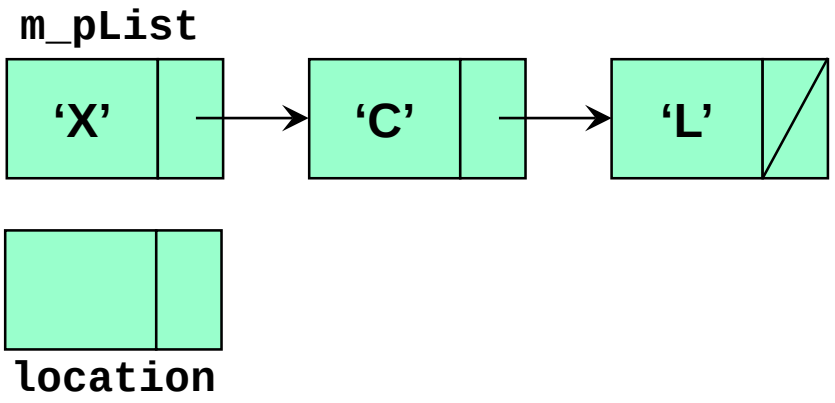
```
newItem = ‘B’;
NodeType<T> *location;
location = new  NodeType<T>;
location->info = newItem;
location->next = m_pList;
m_pList = location
```



Lab04: Reference (1/2)

AddItem ‘B’

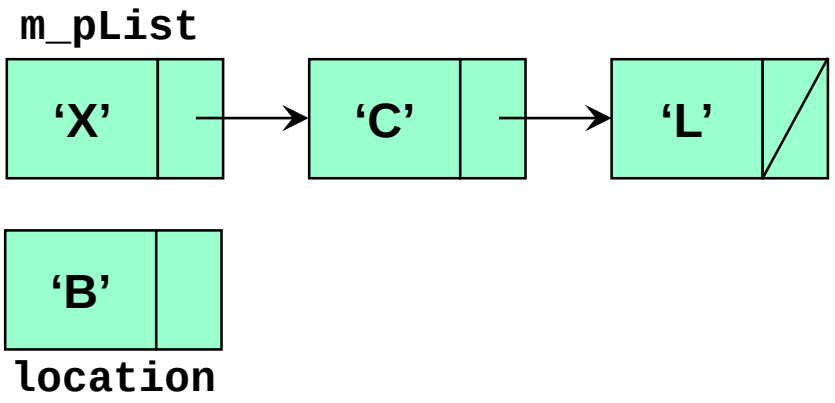
```
newItem = ‘B’;
NodeType<T> *location;
location = new  NodeType<T>;
location->info = newItem;
location->next = m_pList;
m_pList = location
```



Lab04: Reference (1/2)

AddItem ‘B’

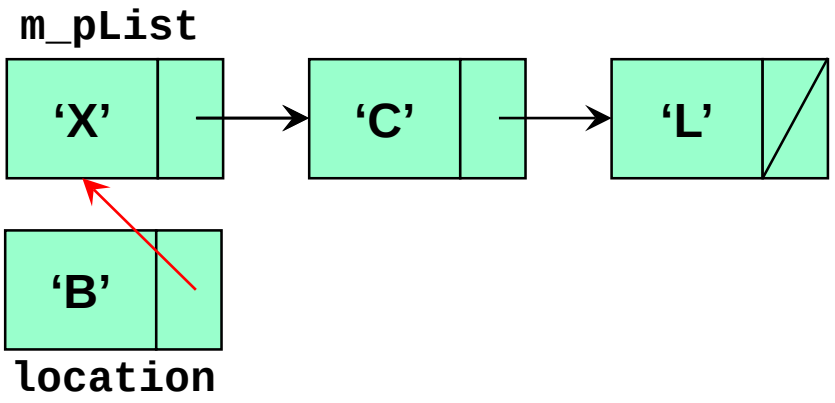
```
newItem = ‘B’;
NodeType<T> *location;
location = new  NodeType<T>;
location->info = newItem;
location->next = m_pList;
m_pList = location
```



Lab04: Reference (1/2)

AddItem ‘B’

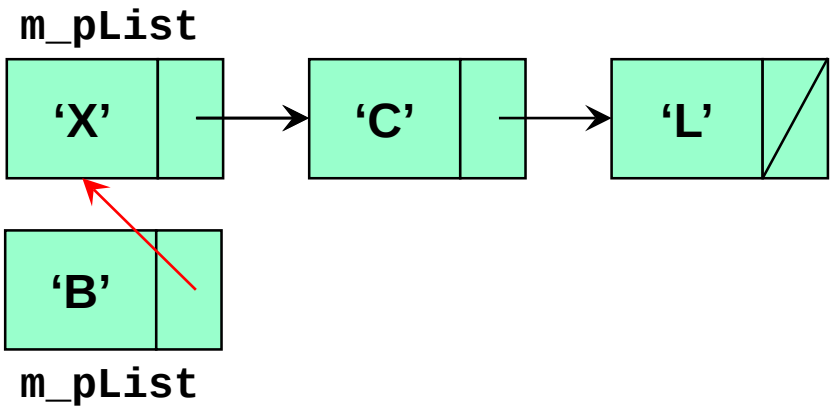
```
newItem = ‘B’;
NodeType<T> *location;
location = new  NodeType<T>;
location->info = newItem;
location->next = m_pList;
m_pList = location
```



Lab04: Reference (1/2)

AddItem ‘B’

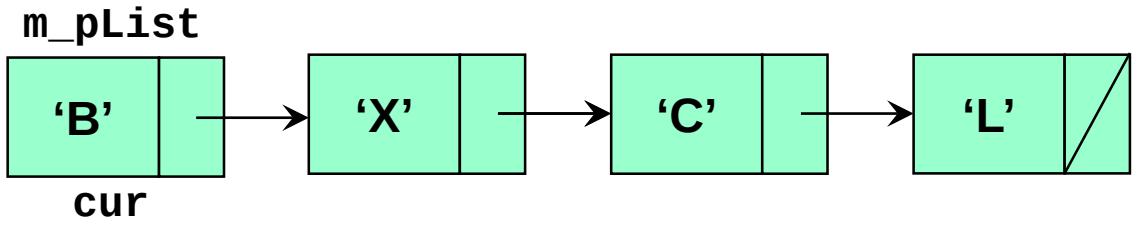
```
newItem = ‘B’;
NodeType<T> *location;
location = new  NodeType<T>;
location->info = newItem;
location->next = m_pList;
m_pList = location
```



Lab04: Reference (2/2)

DeleteItem ‘C’

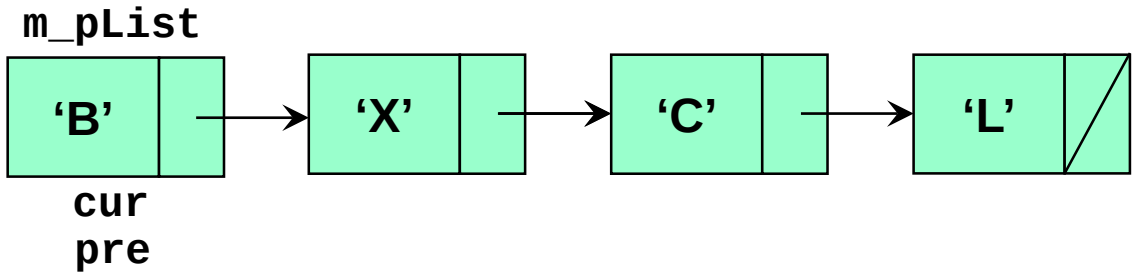
```
cur = m_List;
while(cur != null)
        if(cur->info != DeleteItem->info)
                pre = cur;
                cur = cur->next;
        else
                pre->next = cur->next;
                delete cur;
```



Lab04: Reference (2/2)

DeleteItem ‘C’

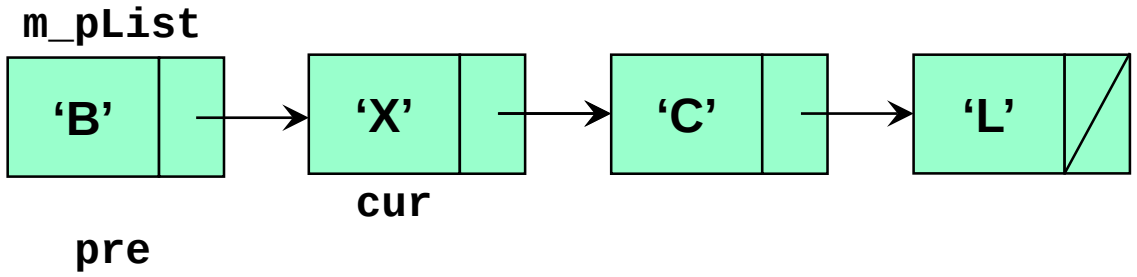
```
        cur = m_List;
        while(cur != null)
                if(cur->info != DeleteItem->info)
                        pre = cur;
                        cur = cur->next;
                else
                        pre->next = cur->next;
                        delete cur;
```



Lab04: Reference (2/2)

DeleteItem ‘C’

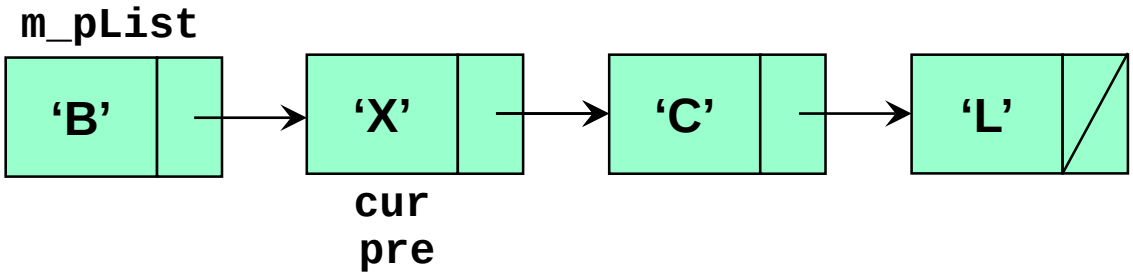
```
cur = m_List;
while(cur != null)
        if(cur->info != DeleteItem->info)
                pre = cur;
        cur = cur->next;
        else
                pre->next = cur->next;
                delete cur;
```



Lab04: Reference (2/2)

DeleteItem ‘C’

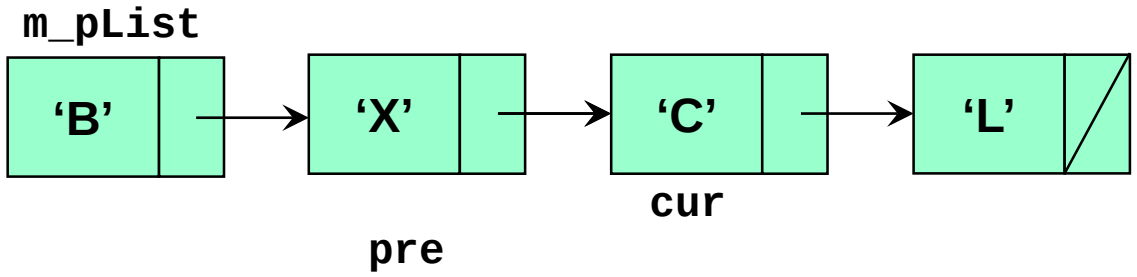
```
        cur = m_List;
        while(cur != null)
                if(cur->info != DeleteItem->info)
                        pre = cur;
                        cur = cur->next;
                else
                        pre->next = cur->next;
                        delete cur;
```



Lab04: Reference (2/2)

DeleteItem ‘C’

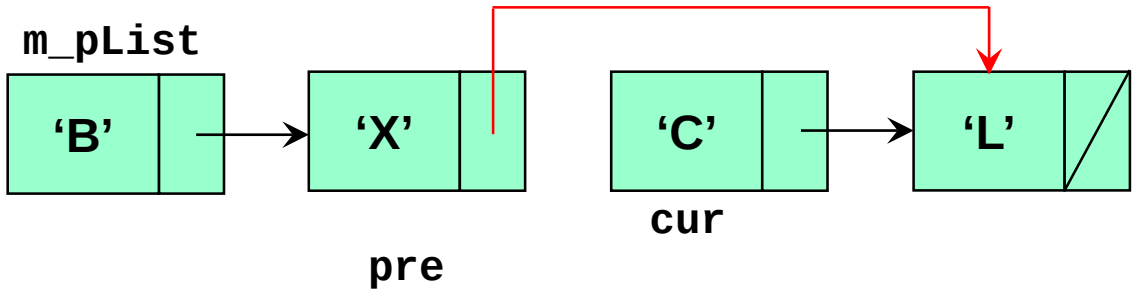
```
        cur = m_List;
        while(cur != null)
                if(cur->info != DeleteItem->info)
                        pre = cur;
                        cur = cur->next;
                else
                        pre->next = cur->next;
                        delete cur;
```



Lab04: Reference (2/2)

DeleteItem ‘C’

```
cur = m_List;
while(cur != null)
        if(cur->info != DeleteItem->info)
                pre = cur;
                cur = cur->next;
        else
                pre->next = cur->next;
                delete cur;
```



Lab04: Reference (2/2)

DeleteItem ‘C’

```
        cur = m_List;
        while(cur != null)
                if(cur->info != DeleteItem->info)
                        pre = cur;
                        cur = cur->next;
                else
                        pre->next = cur->next;
                delete cur;
```