

**CHAPTER 12**

**단순 연결 리스트**

# 실습 1

## 문제 1)

단순 연결 리스트와 메서드들을 구현하시오.  
(메서드 설명은 다음 페이지에)

C:\Users\Oculus\source\repos\ConsoleApplication2\x64\

```
*****
0. Create
1. display
2. Insert Node at beginning
3. Insert Node in specific position
4. Insert Node at end of LinkedList
5. Delete Node at beginning
6. Delete Node at end
7. Delete Node at position
8. ** To exit **
Enter your choice: 0
Enter node data: 12
```

```
*****
0. Create
1. display
2. Insert Node at beginning
3. Insert Node in specific position
4. Insert Node at end of LinkedList
5. Delete Node at beginning
6. Delete Node at end
7. Delete Node at position
8. ** To exit **
Enter your choice: 1
LinkedList: 12
```

```
*****
0. Create
1. display
2. Insert Node at beginning
3. Insert Node in specific position
4. Insert Node at end of LinkedList
5. Delete Node at beginning
6. Delete Node at end
7. Delete Node at position
8. ** To exit **
Enter your choice: 2
Enter node data: 32
```

```
*****
0. Create
1. display
2. Insert Node at beginning
3. Insert Node in specific position
4. Insert Node at end of LinkedList
5. Delete Node at beginning
6. Delete Node at end
7. Delete Node at position
8. ** To exit **
Enter your choice: 1
LinkedList: 32 12
```

```
*****
0. Create
1. display
2. Insert Node at beginning
3. Insert Node in specific position
4. Insert Node at end of LinkedList
5. Delete Node at beginning
6. Delete Node at end
7. Delete Node at position
8. ** To exit **
Enter your choice: 3
Enter node data: 2
Enter position: 2
```

```
*****
0. Create
1. display
2. Insert Node at beginning
3. Insert Node in specific position
4. Insert Node at end of LinkedList
5. Delete Node at beginning
6. Delete Node at end
7. Delete Node at position
8. ** To exit **
Enter your choice: 1
LinkedList: 32 1 2 12
```

# 실습 1

문제 1) 메서드 설명

createLinkedList() // data가 null인 head를 가진 연결 리스트 생성

display() // 현재 연결 리스트에 연결된 node들의 data 출력

insert\_begin() // 연결 리스트의 head에 node 삽입

insert\_end() // 연결 리스트의 tail에 node 삽입

insert\_pos() // position을 입력받고 원하는 position에 node 삽입

delete\_begin() // 연결 리스트의 head 삭제

delete\_end() // 연결 리스트의 tail 삭제

delete\_pos() // position을 입력받고 원하는 position의 node 삭제

## 실습 2

### 문제 2)

문제 1에서 만든 메서드를 활용하여 다음과 같은 리스트를 만드시오.  
[1,2,3,4,5,6,7,8,9,10]의 값을 가지는 연결 리스트를 생성하고  
[3,5,7]의 값을 가지도록 메서드를 호출하시오. (호출 순서는 상관없음)

```
Microsoft Visual Studio 디버그 콘솔
Enter node data: 1
Enter node data: 2
Enter node data: 3
Enter node data: 4
Enter node data: 5
Enter node data: 6
Enter node data: 7
Enter node data: 8
Enter node data: 9
Enter node data: 10
LinkedList: 1 2 3 4 5 6 7 8 9 10
Node Deleted
Node Deleted
Enter node position to delete: 1
Enter node position to delete: 2
LinkedList: 3 5 7
```


## 실습 3

문제 3)

문제 1에서 만든 메서드를 활용하여 다음과 같은 리스트를 만드시오.

[1,2,3], [7,8] 의 값을 가지는 두 개의 연결 리스트를 생성하고

[1,2,3,7,8] 의 값을 가지도록 두 개의 연결 리스트를 연결하는 함수  
concatList() 를 구현하시오.

 Microsoft Visual Studio 디버그 콘솔

```
Create First LinkedList
Enter node data: 1
Enter node data: 2
Enter node data: 3
Create Second LinkedList
Enter the first data: 9
Enter the second data: 10
LinkedList: 1 2 3 9 10
```

## 실습 4

문제 4)

문제 1에서 만든 메서드를 활용하여 다음과 같은 리스트를 만드시오.

[1,2,3,4,5,6]의 값을 가지는 연결 리스트를 생성하고

[6,5,4,3,2,1]의 값을 가지도록 바꾸는

reverseLinkedList() 함수를 구현하시오.

```
Microsoft Visual Studio 디버그 콘솔
Enter node data: 1
Enter node data: 2
Enter node data: 3
Enter node data: 4
Enter node data: 5
Enter node data: 6
LinkedList: 1 2 3 4 5 6
SUCCESSFULLY REVERSED LIST
LinkedList: 6 5 4 3 2 1
```

## 실습 5

### 문제 5)

연결 리스트에 저장된 값이 Palindrome 인지 검사하는  
isPalindrome() 함수를 구현하시오.

Palindrome: 거꾸로 읽어도 제대로 읽는 것과 같은 문장이나 낱말, 숫자, 문자열 등

Microsoft Visual Studio 디버그 콘솔

```
Create LinkedList
Enter node data: 1
Enter node data: 2
Enter node data: 3
Enter node data: 4
LinkedList: 1 2 3 4
The linked list is not a palindrome.
```

Microsoft Visual Studio 디버그 콘솔

```
Create LinkedList
Enter node data: 1
Enter node data: 2
Enter node data: 2
Enter node data: 1
LinkedList: 1 2 2 1
The linked list is a palindrome.
```

# 질문 및 정리

