4주차 결과보고서

전공: 기계공학과 학년: 3학년 학번: 20191820 이름: 김형준

**1. 자료구조**

LinkedList 클래스에 사용된 자료구조는 연결리스트 방식으로 구현된 1차원 큐이다.

연결리스트는 데이터를 저장하는 각각의 노드가 포인터로 처음부터 끝까지 한 줄로 연결되어있는 자료구조를 의미한다.

노드의 포인터는 다음 노드가 있는 경우 다음 노드의 주소값을 저장하고,

없는 경우 NULL값을 저장해 끝을 나타낸다.

큐는 FIFO (First In First Out)의 성질을 만족하며, 데이터를 넣은 순서대로 데이터가 나가는 것을 의미한다.

Stack 클래스에서 사용된 자료구조는 연결리스트 방식으로 구현된 1차원 스택이다.

스택은 LIFO (Last In First Out)의 성질을 만족하며, 마지막으로 넣은 데이터가 첫 번째로 나가는 것을 의미한다.

**2. 알고리즘**

LinkedList, Stack 클래스에 사용되는 노드는 Node 클래스의 인스턴스를 사용한다.

Node 클래스는 template <typename T>를 사용해 템플릿 클래스로 확장되었기 때문에 (파라미터적 다형성 지원) 임의의 자료형을 데이터로 저장할 수 있다. (마찬가지로, 임의의 자료형을 사용하기 위해 LinkedList, Stack 클래스도 템플릿 클래스로 확장됨)

LinkedList 클래스의 first 멤버 변수는 연결리스트의 첫 번째 노드를 가리키며,

current\_size에는 현재 연결리스트에 있는 노드의 개수가 저장된다.

LinkedList 클래스의 메소드 목록 :

- LinkedList () : LinkedList 클래스의 생성자이다. first와 current\_size를 각각 0으로 초기화한다.

- int GetSize() : current\_size의 값을 리턴한다.

- void Insert(T element) : 연결리스트의 맨 앞에 노드를 추가하는 함수이다.

먼저 newnode에 Node를 new 연산자로 생성한 다음 (element를 Node의 data에 저장함)

newnode의 link에 first의 값을 저장해 새로 생성된 노드가 연결리스트의 첫 번째 노드를 가리키게 만들고 first에 newnode의 주소를 저장해 새로 생성된 노드가 첫 번째 노드가 되게 만든다. 마지막으로 current\_size를 1 증가시켜 전체 노드의 개수를 갱신한다.

- virtual bool Delete(T &element) : 연결리스트의 맨 뒤의 노드를 삭제하는 함수이다.

virtual 키워드를 사용해 LinkedList와 LinkedList를 상속받는 클래스의 객체를 가리키는 포인터가 Delete 함수를 호출했을 때 포인터의 자료형이 아닌 객체의 자료형에 맞는 Delete 함수를 실행할 수 있게 했다. (서브타입 다형성 지원)

먼저, first가 0인 경우(노드가 없어서 삭제가 불가능한 경우)에는 false를 리턴해서 함수를 종료한다. 아니면 current에 first, previous에 0을 각각 저장하고 while문을 사용해

마지막 노드에 도달한 다음 (link가 0이면 다음 노드가 없으므로 current에 마지막 노드의 주소가 담김) previous가 NULL이 아니면 previous -> link = current -> link;로 마지막 노드를 연결리스트에서 삭제하고, 아니면 (이전 노드가 없으므로 전체 노드의 개수는 1개임) first = first -> link;로 마지막 노드를 연결리스트에서 삭제한다.

마지막으로 element = current -> data;로 삭제한 노드의 값을 element에 저장한 다음

delete 연산자를 사용해 삭제한 노드의 메모리를 할당 해제한다. 그리고 current\_size를 1 감소시켜 전체 노드의 개수를 갱신한다.

- void Print() : 연결리스트에 있는 모든 노드를 순서대로 출력하는 함수이다.

current\_size가 0이 아닐 때만 출력하며, 연결리스트의 첫 번째 노드부터 마지막 노드까지 차례대로 순회하면서 값을 양식에 맞게 출력한다.

(마지막 노드의 값 출력시 화살표를 출력하지 않는다.)

Stack 클래스의 메소드 목록 :

Stack 클래스는 LinkedList 클래스를 상속받았으므로 클래스 내부에 재정의된 Delete 함수를 제외한 LinkedList 클래스의 모든 멤버 함수와 변수를 그대로 사용한다.

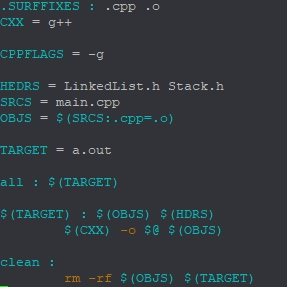
- bool Delete (T &element) : 연결리스트의 맨 앞 노드를 삭제하는 함수이다.

먼저, first가 0인 경우에는 false를 리턴해서 함수를 종료한다. 아니면 current에 first의 값을 저장한 다음 삭제할 첫 번째 노드의 데이터를 element에 저장한다.

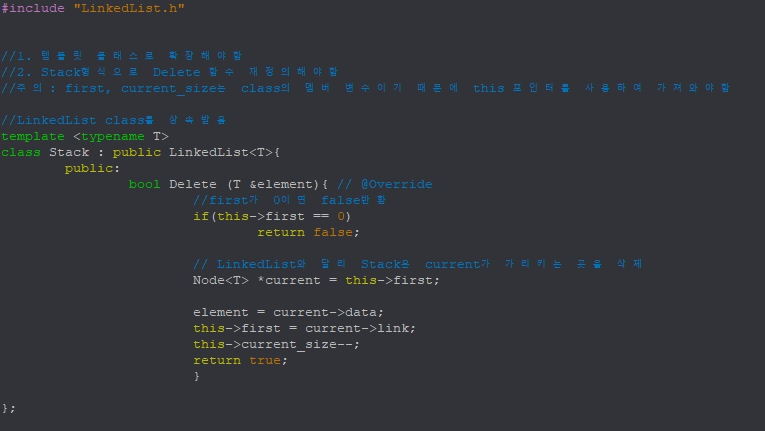
그 다음, first에 다음 노드의 주소값(다음 노드가 없는 경우 NULL)을 저장해서 첫 번째 노드를 연결리스트에서 삭제하고 current\_size를 1 감소시켜 전체 노드의 개수를 갱신한다.

**3. 실습 코드**

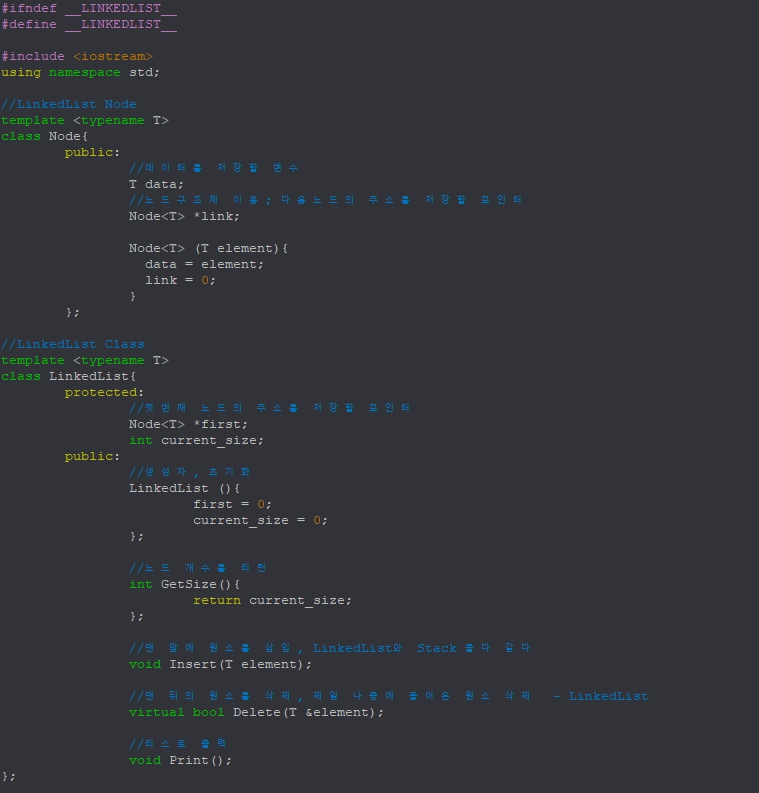
3-1. Makefile



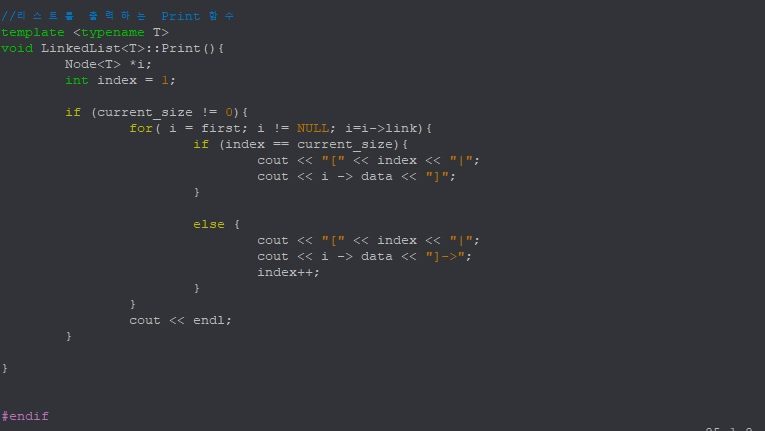
3-2. Stack.h



3-3. LinkedList.h







3-4. main.cpp

