4주차 결과보고서

전공: 아트엔테크놀로지 학년: 3학년 학번: 20191172 이름: 함승우

1.

LinkedList.h에서 구현된 LinkedList class를 템플릿 class로 구현하고, 이 LinkedList class를 상속받아 Stack.h의 Stack을 구현하는 것이 실습 과제 였다. 하나하나씩 설명을 하자면,

LinkedList.h에는 크게 두 가지의 클래스로 구성이 되어있는데 첫번째는 Node class이다. 이 Node class는 다음 노드의 주소를 저장할 포인터이다, 자료형에 구애받지 않기 위해 template을 사용을 하는데, template<typename T>>를 사용해 입력받은 data와 Node를 T 자료형으로 바꿔줘야 한다.

LinkedList class에는 아래와 같은 기능을 한다.

LinkedList를 생성하며, 첫번째 노드를 0으로, 개수를 0으로 초기화하고, Getsize()는 노드 개수를 리턴하며, 멤버함수로는 void Insert(T element);, virtual bool Delete(T & element);, void Print();의 함수가 있다.

Insert(T element) 함수는 새로운 Node(newnode)를 생성하고, 이 새로운 노드가 첫번째 노드를 가리키게/newnod -> link = first//fist = newnode//만들고 변수 current\_size를 1만큼 증가시켜 줘서 삽입됨을 알린다.

Delete(T &element)는 첫번째 변수가 first == 0일 경우, 즉 삭제할 부분이 없을 경우 false를 리턴하고, current에 first(첫번째 변수)를 할당하고, while문을 사용하여 반복적으로 마지막 노드까지(current->link == 0)까지 찾아가는 반복문을 사용한다. 이후 current를 삭제해 delete 함수를 완성시킨다.

print()함수는 현재 LinkedList 안에 있는 모든 data를 출력을 하는데, 마지막(index == current\_size)까지 출력을 한다. 마지막이 아니라면 ([X|Y]->)의 형태로 출력하, 마지막이라면 ([X|Y])의 형태로 출력하게끔 한다.

Stack.h에는 LinkedList 클래스를 상속받고, 이 LinkedList의 삭제 순서를 바꾸면 되기 떄문에 delete 함수만을 개조했다. Linked List는 마지막 노드까지 찾고 삭제를 하지만, 이와 달리 current가 가리키는 값을 삭제한다. 따라서

element = this->first->data;

Node<T> \*current = this->first;

this->first = this->first->link;

delete current;

this->current\_size--;

return true;

의 코드로 구현할 수 있다.

마지막으로 main.cpp의 경우 처음 preMenu를 출력해 사용자가 LinkedList로 할지, Stack으로 할지 결정하게 하고, 이후 1[삽입], 2[삭제], 3[출력] 4[종료]를 입력하게 한다.

각각 insert(), delete(), getSize()함수를 이용한다. User가 이전에 입력한 1이냐 다른 정수인가에 따라 Stack을 형성하고, LinkedList를 생성하기 때문에 2[삭제] 부분에서 delete가 일어난다.