4주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20212022 이름: 이예준

**1.**

**LinkedList 클래스**

클래스의 다형성을 위해 템플릿을 선언했다. template<class T>를 클래스 앞에 선언하여

기존 int 자료형만 가능했던 클래스가 다른 자료형에도 적용이 가능해졌다.

함수도 void LinkedList<T>::Insert(T element)처럼 이름 앞에 템플릿<T>를 붙이고,

인자의 자료형도 T를 사용하여 선언했다. 객체 Node도 Node<T>\*first처럼 템플릿<T>를 붙였다.

**Stack 클래스**

Stack 클래스는 class Stack : public LinkedList<T>처럼 LinkedList 클래스를 상속받아서 구현했고

Stack 클래스도 다형성을 위해 template<class T>를 클래스 앞에 선언했다.

Stack의 특성 후입선출을 의해 Delete 함수는 LinkedList처럼 마지막 원소가 삭제되는 것 아니라

맨 앞에 있는 원소가 삭제되야 되기 때문에 함수를 재정의해야 한다.

따라서 virtual bool Delete (T &element)로 함수를 Overriding한다.

Stack의 멤버변수를 사용하는지 LinkedList의 멤버변수를 사용하는지 모호하기 때문에

함수 내에서 Stack의 멤버변수에 접근하기 위해서 this를 사용한다.

만약 Stack의 원소가 없다면 함수를 중지하기 위해 this->first가 없을 때 false를 반환한다.

Node<T> \* current는 맨 앞에 있는 원소를 가리키고, Node<T> \* next는 current가 가리키고 있는

원소 다음에 있는 원소를 가리킨다.그 다음 첫 요소를 가리켜야 하는 first가 next를 가리키게

한다. 그리고 제거해야 하는 원소의 data를 element에 저장하고, delete 명령어를 써서

current 원소를 제거한다. 마지막으로 this로 원소의 개수 current를 가리킨 뒤 노드의 개수 1개를

뺀다.

**최대 크기**

main.cpp 파일에서 #define MAX\_SIZE 5 같이 최대 크기를 임의로 정의해준다.

그 다음 삽입 명령어 1을 입력 받았을 때, p->Getsize()로 리스트의 원소 개수를 받아온다.

원소의 개수와 MAX\_SIZE를 비교하여 같거나 크면 “리스트의 최대 크기에 도달했습니다”

에러 매세지를 화면에 출력하고 원소 삽입과정을 중단한다.