4주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20221532 이름: 김민기

**1.**

i) LinkedList를 템플릿기반 클래스르 변환. 템플릿을 사용하는 것은 data와 element, Node, first이므로 이들의 타입을 변경한다.

ii) Stack 클래스는 LinkedList를 상속받아 delete 함수만 재정의 해준다. 맨 앞의 원소를 제거해야 하므로 Stack이 비어 있을 때가 아니면 first의 값을 두번째 값으로 변경한 후 first의 값을 제거한다. 한 원소를 제거했으므로 current\_size도 1감소시킨다. Stack 이 비어 있을 때엔 어떤 행동을 하지 않고 delete 함수를 종료한다.

template <typename T>

class Node{

public:

T data; //데이터를 저장할 변수, 템플릿 T로 변경

Node<T> \*link; //노드 구조체, 다음 노드의 주소 저장, Node는 템플릿을 사용하 므로 <T>명시

Node(T element){ // 생성자, 인자를 템플릿으로 받음

data = element;

link = 0;

}

};

//LinkedList Class

template <typename T>

class LinkedList{

protected:

Node<T> \*first; //첫번째 주소 담을 포인터, 템플릿 사용 명시

int current\_size;

public:

LinkedList(){ //생성자

first = 0;

current\_size = 0;

};

int GetSize(){ //배열 크기 리턴

return current\_size;

};

void Insert(T element); //리스트 맨 앞에 원소 저장

//새 Node생성 후, first로 설정.

virtual bool Delete(T &element); //리스트 맨 끝의 원소 제거.

//리스트의 끝으로 접근하여 제거.

void Print(); //리스트 출력

};

//Stack Class

template <typename T>

class Stack : public LinkedList<T>{ //LinkedList 상속받음

public:

bool Delete (T &element); //LinkedList의 delete 재정의. 리스트의 첫번째 원소 제거. First의 값을 두번째 원소로 변경 후 첫번째 원소 제거.

}