

4주차 결과보고서

전공: 컴퓨터공학과

학년: 2학년

학번: 20231632

이름: Jumagul

Alua

1. 프로그램 구조 및 목적

이 프로그램은 입력자한테 2 개의 Stack 이나 Linked List이나 프로그램 선택을 받는다. 만약 1이라면 Stack을 동작하는 프로그램으로 계속한다. 만약 다른 숫자가 입력 되면 Linked list으로 설정할 것이다. 그 다음 main 함수에서 4개 메뉴 중 선택을 하라는 출력이 나온다.

1. 삽입: 요소를 받아서 삽입하는 삽입 선택.
2. 삭제: 삽입한 요소를 삭제할 삭제 선택.
3. 출력: 삽입과 삭제된 데이터를 반환을 하는 출력 선택.
4. 종료: 마지막으로 프로그램 종료를 하는 종료 선택.

Stack과 LinkedList를 위한 삽입과, 특히 삭제 부분이 다른 원칙으로 동작을 한다. 먼저, 스택은 데이터를 마지막에 삽입한 것이 먼저 삭제되는 Last In, First Out 원칙을 따라한다. 반대로, 연결 리스트는 먼저 삽입한 데이터를 삭제하는 원칙으로 동작한다.

2. LinkedList.h 함수:

이 함수에는 주어진 소스코드 기본 클래스를 템플릿 클래스로 반환하는 것이다. 예를 들면, 주어진 코드에 있는 이러한 부분이 템플릿으로 바꿀 수 있다.

A) `Int data; -> T data, int같은 type을 T로 반환할 것.`

- B) Node 뒤에 <T> 를 넣을 것.
- C) 함수 시작할때 앞에 template<class T> 를 넣을 것.
- D) <LinkedList>뒤에 <T> 를 넣을 것.

3. Stack.h 함수:

- “*Bool Delete(T &element)*” T 타입의 요소를 삭제한다. Element 변수에 저장된다.
- “*if(this->first==0){*

return false;

}” 스택이 비어 있는지 확인한다.
- “*Node<T> *current = this->first;*” 스택에서 삭제할 요소를 찾기 위해 현재 첫 번째 노드를 가리키는 포인터를 생성한다.
- “*this->first=this->first->link;*” 첫 번째 노드를 삭제한다.
- “*element=current->data;*” 삭제된 데이터를 element 변수에 저장한다.
- “*delete current;*” 메모리를 해제한다.
- “*this->current_size--;*” 스택의 크기를 감소시킨다.
- “*return true;*” 요소가 삭제되었으므로 true를 반환한다.