전공: 국제한국학 학년: 4 학번: 20181202 이름 : 김수미

1. 실험 시간에 작성한 프로그램의 알고리즘과 자료구조를 요약하여 기술하시오

이번 실습에서는 객체지향 프로그래밍의 상속과 다형성 기능을 활용하여 연결리스트와 스택 구조를 구현하는 프로그램을 작성해보았다. 프로그램은 Node 클래스, LinkedList 클래스, Stack 클래스, main.cpp 파일로 구성되며 아래는 각각의 파일에 대한 설명이다.

1) Node 클래스

데이터필드와 링크 필드로 구성되는 Node 클래스를 이용하여 LinkedList 자료구조를 구현했다. LinkedList를 이루는 각각의 노드는 data 필드에 템플릿 자료형의 데이터를 저장할 수 있으며, link 필드는 다음에 연결되는 노드의 주소를 가리킴으로써 전체적으로 하나로 연결된 LinkedList의 형태를 형성할 수 있게 해준다.

2) LinkedList 클래스

위의 노드 클래스 변수와 정수형 변수 current_size를 가지는 LinkedList 클래스는 현재 생성되어있는 노드 개수를 리턴하는 GetSize, 리스트의 맨 앞에 원소를 삽입하는 Insert. 맨 뒤의 원소(제일 나중에 들어온 원소)를 삭제하는 Delete, 현재 리스트를 출력하는 Print 멤버함수로 구성되어 있다. 노트포인터 변수 first는 언제나 리스트의 가장 첫번째 (가장 나중에 삽입된) 원소를 가리킨다.

- GetSize : current_size 변수에 저장되어있는 값을 return
- Insert : 새로운 노드포인터 생성, 기존 리스트 맨 앞 노드의 주소를 가리키도록 설정
- Delete : first 노드부터, 더이상 뒤에 연결된 것이 없는 마지막 노드를 순차적으로 찾아 그것을 삭제한다.
- Print : first 노드부터 더이상 뒤에 연결된 것이 없는 마지막 노드까지 차례대로 읽으며 data 필드에 저장된 값과 몇번째 원소인지(index)를 출력한다.

2) Stack 클래스

LinkedList 클래스를 상속받으며 Delete 부분만 재정의되었다.
LinkedList 클래스에서는 마지막 노드를 찾아 그것을 Delete 했지만, Stack 클래스에서는 first가 가리키는 원소를 삭제하고, first는 기존에 가장 첫번째 원소에 연결되어 있던 원소를 가리키도록 바꾼다(첫번째 원소가 삭제됨에 따라 두번째 원소가 이제 첫번째가 됨)

3) main.cpp 파일

헤더파일에서 계산된 것들을 받아와 보기 좋게 형식에 맞춰 출력할 수 있도록 해주는 내용을 담고 있다.