

# 2021 Spring OOP Assignment Report

과제 번호 : Programming Assignment #4

학번 : 20200437

이름 : 김채현

Povis ID : kch3481

## 명예서약 (Honor Code)

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

## [문제 1번]

### 1. 프로그램 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- Set STL의 일부와 비슷한 기능을 하는 MySet Class를 구현한다.
- 각 S#.txt 파일과 E#.txt 파일에서 Set, Element의 값을 읽어와 input\_p1.txt 파일에 있는 계산을 하여 output\_p1\_20200437.txt 파일에 출력한다.

### 2. 프로그램 구조 및 알고리즘

#### 1) MySet Class

MySet Class는 Set의 각 원소를 data로 가진 Node를 Linked List로 저장, 관리한다. LinkedList에서 맨 처음 node와 맨 마지막 node는 유의미한 값을 가지지 않고 node를 이어주는 역할만 하는 일명 dummy node이다. 따라서 list가 비었다는 의미는 맨 처음 node와 맨 마지막 node만 존재하고 데이터를 가진 node는 존재하지 않는다는 의미로 구현하였다.

#### (1) Node

먼저 LinkedList의 node가 되는 struct Node는

- ① 이전 노드를 가리키는 pointer NodeType \*prev
- ② Set의 원소를 저장하는 int data
- ③ 다음 노드를 가리키는 pointer NodeType \*next

로 이루어져 있다.

#### (2) member

list의 맨 처음 노드를 가리키는 포인터 Node \*pFirst, list의 맨 마지막 노드를 가리키는 포인터 Node \*pLast, 리스트에 저장된 node의 수(dummy node 제외) int length가 존재한다.

### (3) method

#### ① 생성자와 소멸자, 복사생성자

리스트 생성시 맨 처음과 마지막에 들어가는 dummy node가 생성되어야 한다. 그렇기에 pFirst와 pLast에 동적할당을 통해 NodeType을 생성해주고 두 노드를 이어준다. length는 0으로 저장해준다. 초기값이 있는 생성자는 추가로 parameter로서 int\* array와 int S\_number를 받아와 array가 가리키는 동적할당 배열에 있는 data를 하나씩 리스트에 추가해준다. 복사생성자도 존재한다. 소멸 시에도 동적할당 해준 pFirst와 pLast를 해제해준다.

#### ② operator overloading

각 operator에 맞게 LinkedList의 Node를 추가, 삭제해준다. operator= 의 경우 return type을 MySet&로 변경해주었다.

#### ③ contains, isEmpty, getSize

contains의 경우 LinkedList에 해당 data를 가진 Node들이 있는지 확인한 후 bool type으로 return한다. isEmpty의 경우 length가 0이면 true를 return한다. getSize는 length를 return한다.

#### ④ toString

LinkedList의 Node에 있는 data를 string으로 저장하여 return한다.

## 2) main 함수

### (1) save\_Set

S#.txt 파일에서 내용을 읽어와 MySet을 생성하고 Set\_array에 저장한다. main함수에서 동적할당 할 MySet\* Set\_array를 생성한 후 parameter로 받아온다. int i=1부터 반복문을 돌면서 S(i).txt 파일이 있는지 열어 파일의 개수 S\_number를 구한다. 파일의 개수만큼 Set\_array를 동적할당 해주고 순서대로 S#.txt 파일의 내용을 읽어와 임시로 int array에 저장한다. array와 원소의 개수를 parameter로 넘겨주어 Set을 생성하고 Set\_array에 저장해준다.

### (2) save\_Element

save\_Set과 비슷한 기능을 한다. main함수에서 동적할당 할 int\* Element\_array를 생성한 후 parameter로 받아온다. int i=1부터 반복문을 돌면서 E(i).txt 파일이 있는지 열어 파일의 개수 E\_number를 구한다. 파일의 개수만큼 Element\_array를 동적할당 해주고 순서대로 E#.txt 파일을 Element\_array에 저장해준다.

### (3) find\_number

parameter로 넘겨받은 string을 첫 글자를 떼고 숫자로 바꾸어 return해준다. S1, E1과 같은 string에서 S, E를 떼고 1만 return해주는 식으로 사용된다.

### (4) change\_Sfile

parameter로 number와 MySet을 넘겨받아 number에 해당하는 S#.txt 파일을 열어 MySet의

내용으로 바꾼다.

#### (5) 이외의 main함수 알고리즘

S#.txt파일과 E#.txt파일로부터 내용을 불러와 각각 Set\_array, Element\_array에 저장해준 후 input\_p1.txt 파일을 열어 한 줄씩 string으로 읽어온 후 stringstream을 이용해 각각 string으로 받아온다. operation에 맞게 연산을 진행한 후 결과값을 output\_p1\_20200437.txt파일에 적어준다. =operation의 경우 해당 S#.txt 안의 내용도 바꾸어야 하기 때문에 change\_Sfile 함수를 호출하여 파일 안의 내용을 새로 출력해준다.

### 3) 토론 및 개선

본 프로그램은 Set STL의 일부를 구현하여 입력파일에 있는 연산을 수행하여 출력파일로 출력하는 프로그램이다. Class MySet의 구현자체는 어렵지 않았다. 하지만 매번 쓰던 LinkedList로 구현하여 data를 관리하였다는 점이 아쉽다. 이번에는 STL container가 금지되었지만 다음에는 다양한 container를 써보도록 노력할 것이다. 또한 아쉬운 점은 입력파일에서 S1 S2와 같이 string을 받아온 후 해당 string에 해당하는 txt 파일을 여는 방식으로 구현하였다면 좋았을 것이라는 생각이 든다. 물론 본 과제에서는 error 핸들링을 고려하지 않기 때문에 S1부터 순서대로 해당파일이 없을 때까지 열어도 알고리즘 시간 상 차이는 많이 없을 것이라 생각되지만 같은 방식으로 한 결과 2번 과제에서는 문제가 발생하였다.

## [문제 2번]

### 1. 프로그램 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- Matrix STL의 일부와 비슷한 기능을 하는 MyMatrix Class를 구현한다.
- 각 M#.txt 파일에서 Matrix의 값을 읽어와 input\_p2.txt 파일에 있는 계산을 하여 output\_p2\_20200437.txt 파일에 출력한다.

### 2. 프로그램 구조 및 알고리즘

#### 1) MyMatrix Class

##### (1) member

2차원 배열을 가리키는 이중포인터 T\*\* m, 행과 열의 길이 int N, 파일의 번호(M#.txt에서 #) int file\_number, 정사각형 아님 여부를 알려주는 int Not\_Square, 원소빠짐 여부를 알려주는 int Lost\_Element.

## (2) method

### ① 생성자와 복사생성자, 소멸자

초기화 값이 없는 생성자에서는  $N$ 을 1로 설정해주고 0인 MyMatrix를 만들어준다. parameter가 int n형 하나인 생성자는  $N$ 에  $n$ 을 저장하고 크기가  $N*N$ 인 2차원 배열을 생성해준다. parameter가 int n, T\*\* m\_array 두 개인 생성자는  $n$ 에  $N$ 을 저장하고 이미 동적할당 되어 생성된 m\_array를  $m$ 에 저장한다. 복사생성자와 소멸자도 존재한다. 소멸자에서는 동적할당 된 배열을 모두 해제해준다.

### ② operator overloading

각 operator에 맞게 2차원 배열에 연산을 진행해준다. operator=의 경우 return type을 MyMatrix&로 바꾸주었다.

### ③ transpose

transpose 행렬특성에 맞게 2차원 배열의 값을 바꿔준다.

### ④ toString

LinkedList의 Node에 있는 data를 string으로 저장하여 return한다.

### ⑤ getter, setter

필요하여 추가로 구현하였다.

## 2) main 함수

### (1) save\_Matrix

M#.txt 파일에서 내용을 읽어와 MyMatrix을 생성하고 Matrix\_array에 저장한다. main함수에서 동적할당 할 MyMatrix\* Matrix\_array를 생성한 후 parameter로 받아온다. int i=1부터 반복문을 돌면서 M(i).txt 파일이 있는지 열어 파일의 개수 Matrix\_number를 구한다. Matrix\_number만큼 임의로 M#.txt에서 #을 담은 배열 int\* M\_file\_number를 선언하고 순서대로 저장해준다. Matrix\_array를 동적할당 해주고 순서대로 M#.txt 파일의 내용을 읽어와 총 행의 크기와 각 행의 크기를 계산하여 NOT\_SQUARE인지 LOST\_ELEMENT인지 체크해준다. 아니라면 MyMatrix를 생성하고 Matrix\_array에 저장해준다. M\_file\_number를 참조하여 MyMatrix의 file\_number도 set해준다.

### (2) find\_number

1번에서의 find\_number 함수와 같은 역할을 하는 함수이다. 단, 차이점은 string에서 받은 숫자의 file\_number를 가진 MyMatrix의 배열번호를 return해준다는 것이다.

### (3) change\_Mfile

1번에서의 change\_Sfile 함수와 같은 역할을 하는 함수이다. M#.txt 파일의 내용을 바꾸는 함수이다.

#### (4) 이외의 main함수 알고리즘

M#.txt파일로부터 내용을 불러와 Matrix\_array에 저장해준 후 input\_p2.txt 파일을 열어 한 줄씩 string으로 읽어온 후 stringstream을 이용해 각각 string으로 받아온다. operation에 맞게 연산을 진행한 후 결과값을 output\_p2\_20200437.txt파일에 적어준다. =operation의 경우 해당 M#.txt 안의 내용도 바꾸어야 하기 때문에 change\_Mfile 함수를 호출하여 파일 안의 내용을 새로 출력해준다. 전체적인 알고리즘은 1번과 비슷하다.

### 3) 토론 및 개선

본 프로그램은 Matrix STL의 일부를 구현하여 입력파일에 있는 연산을 수행하여 출력파일로 출력하는 프로그램이다. STL이 허용되었는데 잘 사용하지 못한 것 같아 아쉽다. 또한 아쉬운 점은 입력파일에서 M1 M2와 같이 string을 받아온 후 해당 string에 해당하는 txt 파일을 여는 방식으로 구현하였다면 좋았겠지만 반복문을 이용하여 M(i).txt를 일일이 다 열어보는 방식으로 코드를 구현하여 **컴파일에 굉장히 많은 시간이 걸린다(약 15분정도)**. 조교님께 문의한 결과 1000000개의 파일을 열 수 있다면 충분하다는 답변을 받아 M1000000까지의 파일을 열 수 있도록 구현하였다. 1번과 다르게 본 문제에서는 중간에 빠지는 M#.txt 파일이 있을 수 있기 때문에 open되지 않는 파일이 있다고 해서 반복문을 빠져나갈 수 없어서 어쩔 수 없이 많은 시간이 소요된다. 다음에는 알고리즘의 시간까지 고려해서 최적화된 코드로 구현하고 싶다.