



# Object-Oriented Programming Report

Assignment 2-3

Professor	Donggyu Sim
Department	Computer engineering
Student ID	2022202065
Name	박나림
Class (Design / Laboratory)	2 / C
Submission Date	2023. 4. 27

## Program 1

### □ 문제 설명

직원 정보를 관리하는 프로그램으로, Employee 클래스로 구현하는 문제이다. 이름, 나이, 국가, 직업을 정보로 가지며 사용자가 insert 를 입력하면 클래스 배열에 각각 정보 추가를, find 를 입력하면 이름을 이용해 검색하여 일치하는 정보를 출력하며 change 입력 시 이름을 통해 일치하는 정보의 내용들을 수정한다. print 입력 시에는 클래스 배열의 총 정보들을 출력하며, exit 를 입력하면 프로그램을 종료시킨다. 이때 직원 정보들은 최대 10 명까지이다.

#### -구현 방법

먼저 클래스에서, 멤버함수로 위 정보들 4 개를 가지도록 하며 멤버함수로는 입력 받은 정보들을 매개변수로 가지는 생성자, 이름 일치 검사용의 bool 형 함수, void 형으로 print 함수와 change 함수를 선언해준다. 생성자에서 정보들을 초기화 시켜줄 때에는, 정수타입인 나이는 상관 없지만 포인터인 문자열들은 먼저 동적할당을 해주고 strcpy 를 통해 복사하여 초기화 시킨다. 이때 변수 이름 구분을 위해 this 포인터를 사용하여 쓴다. 이름 일치 검사 함수에서는 검색할 이름을 받아 strcmp 를 이용하여 일치할 시 1, 그렇지 않을 시 0 을 반환하도록 만든다. 정보 출력 함수에서는 정보들을 출력하며, 정보 수정 함수에서는 수정할 정보들을 매개변수로 받아서 복사시킨다. 그 전의 정보 삽입과 달리 수정에서는 공간을 그대로 쓰므로 따로 동적할당 없이 복사만 해준다.

main 함수 내에서는 먼저 클래스 객체 배열로 10 칸을 생성하고, 전체 while 문으로 돌면서 사용자로부터 명령어 문자열을 입력 받으면 if 문으로 검사하여 각 맞는 동작을 수행하도록 시킨다. insert 이면 정보들을 입력 받아서 클래스 객체 배열을 0 부터 증가시키면서 생성자로 정보들을 전달해준다. print 일 경우엔 for 문으로 0 번째 칸부터 돌면서 클래스 객체 배열들을 모두 출력시켜준다. find 에서는 먼저 검색할 이름을 입력 받아서, 이름 일치 함수를 이용하여 차례대로 검사하다가 일치할 때 해당 객체 배열 값을 출력시켜준다. 이때 불일치 경우에 불일치 변수를 증가시켜서 마지막에 그 변수의 값과 반복 횟수 값이 같으면 검색 결과가 없음을 출력해준다. change 에서도 마찬가지로 검색한 이름과 일치하는 정보를 찾아서 정보들을 수정해주며, exit 에서는 while 문을 끝내며 프로그램을 종료시키도록 구현한다.

## □ 결과 화면

```
Microsoft Visual Studio 디버그 콘솔
insert John 23 USA designer
insert James 28 Korea developer
insert Jessica 27 Japan manager
print
=====print=====
Name: John
Age: 23
Country: USA
Job: designer
=====
Name: James
Age: 28
Country: Korea
Job: developer
=====
Name: Jessica
Age: 27
Country: Japan
Job: manager
=====
find James
=====find=====
Name: James
Age: 28
Country: Korea
Job: developer
=====
change James Jason 22 China writer
print
=====print=====
Name: John
Age: 23
Country: USA
Job: designer
=====
Name: Jason
Age: 22
Country: China
Job: writer
=====
Name: Jessica
Age: 27
Country: Japan
Job: manager
=====
exit
C:\Users\박나림\OneDrive\바탕 화면\과제\객체
```

insert, print, find, change, exit 의 명령어들을 입력하면 각각 그에 맞는 동작을 수행하여서 작동되는 모습이다. (\*이어진 화면)

## □ 고찰

처음에 정보들을 배열로써 따로 저장하는 방법이 바로 안 떠올라서 여러 시행착오들을 겪었다. 정보들을 묶는 class 를 하나 더 만들었다가, 다른 함수들의 사용이 불편해져서 main 함수내에서 객체 배열을 생성하는 방법으로 수정했다. 그러면서 처음에 insert 에서 정보들을 클래스로 넘길 때 객체 배열 생성과 동시에 넘기려는 방식으로 하다가 오류가 나서 따로 식을 세우는 방식으로 수정하기도 했다. 또한 제일 시행착오를 겪었던 건 문자열 초기화 부분이었는데, 처음에는 정수형인 나이 값 초기화처럼 단순히 this->name=name; 식으로 했다가 새로운 값이 저장이 안되는 상황이 발생하였다. 그래서 이 부분에 대해 더 공부를 하여, 먼저 동적할당으로 칸을 더 만든 다음에 값을 저장해줘야 함을 깨달았다.

## Program 2

### □ 문제 설명

학생 정보를 저장하는 클래스를 만들고, 그 클래스 객체 배열을 갖는 학교 클래스를 구현하여 사용자로부터 명령어들을 입력 받으면 그에 맞는 동작을 수행되게 하는 프로그램이다. 총 5 가지의 명령어로, 먼저 `new_student` 를 입력 받을 시 학생의 이름, 나이, 분반명을 저장한다. `sort_by_name` 은 각 학생 배열에서 이름순으로 다시 정렬하여 저장하는 역할이다. 이때 편한 정렬을 위해 학생 이름은 소문자로만 받는 것으로 가정한다. `print_all` 은 모든 배열을 다 출력하며, `print_class` 는 사용자로부터 추가로 검색할 분반명을 입력 받아서 일치하는 반의 학생들의 정보들을 출력하고 총 그 반의 학생 수까지 같이 출력해주는 명령어이다. 마지막으로 `exit` 를 입력하면 프로그램을 종료시킨다. 이러한 과정들을 구현할 때, 학생 클래스는 학교 클래스 외에 바깥에서 접근할 수 없도록 해야 한다.

#### -구현 방법

먼저 학생 클래스와 학교 클래스를 만들도록 한다. 학생 클래스에서는 학생들의 정보를 멤버 변수로 가지며, 그 외 멤버 함수로는 생성자들과 각 객체에서 정보 값을 반환할 `get` 함수를 만들어준다. 학교 클래스는 그러한 학생 클래스 타입의 배열과 `size` 를 멤버 변수로 가지며, 멤버 함수로는 `exit` 제외 각 명령어 함수들과 `size` 전달용 함수, 분반 이름 검색 함수를 만들어 준다. `size` 는 `private` 으로 선언되어 있기 때문에 `main` 함수에서는 접근을 못하므로 따로 `public` 에 함수를 만들어 준 것이다. 그래서 분반 검색을 할 때 해당 `size` 만큼 검사할 수 있도록 한다. 대부분의 함수들은 1 번째 문제와 비슷한 원리로 구현하지만, 이번 문제에서는 따로 클래스 객체 배열을 저장한 학교 클래스가 있으므로 해당 배열에서 `->` 을 이용하여 `get` 정보함수들을 사용하는 형식으로 구현한다. 또한 정렬은 선택 정렬 방법을 사용하여, 이전 배열 값과 다음 배열 값의 문자열 크기를 비교하며 교환하는 형식으로 한다. 문자열의 사전 순 정렬을 위해 `string` 을 이용하여 따로 저장한 다음 비교하는 방법으로 한다. 이러한 명령어들은 `main` 함수 내에서 `if` 문을 통해 각 명령을 수행하게 하며, 마지막으로 `exit` 를 입력 받을 시 전체 `while` 문을 벗어나며 프로그램을 종료시키는 것으로 한다.

## □ 결과 화면

```
Microsoft Visual Studio 디버그 콘솔
new_student Jim 8 Hope
new_student Alice 9 Love
new_student Claude 8 Hope
new_student Megan 11 Wish
print_all
====print_all====
Name: Jim
Age: 8
Class: Hope
-----
Name: Alice
Age: 9
Class: Love
-----
Name: Claude
Age: 8
Class: Hope
-----
Name: Megan
Age: 11
Class: Wish
-----

print_class Hope
====print_class====
Name: Jim
Age: 8
Class: Hope
-----
Name: Claude
Age: 8
Class: Hope
-----
Number of classmates : 2

sort_by_name
print_all
====print_all====
Name: Alice
Age: 9
Class: Love
-----
Name: Claude
Age: 8
Class: Hope
-----
Name: Jim
Age: 8
Class: Hope
-----
Name: Megan
Age: 11
Class: Wish
-----
exit
C:\Users\박나림\OneDrive\바탕 화면\과제\2
```

new\_student, print\_all, print\_class, sort\_by\_name, exit 의 명령어들을 입력하면 각각 그에 맞는 동작을 수행하여서 작동되는 모습이다. (\*이어진 화면)

## □ 고찰

이번 문제에서는 그 전 문제와 달리 객체 배열을 저장할 class 를 따로 선언해주는 방식이라 그에 맞는 방법을 생각하는 게 조금 까다로웠다. 특히 학생 클래스는 main 함수에서 접근할 수 없도록 하였으므로, 명령어들을 수행할 함수들을 그 전과 다르게 전부 학교 클래스에 만들었다. 그러자 학생들의 정보 값들을 이용할 때 단순히 지역 변수로 하면 학교 클래스에서는 그 변수가 없으므로, 따로 get 함수를 만들어주어서 값을 불러오는 형식으로 바꾸었다. 또한 가장 시행착오를 겪었던 부분은 정렬 부분이었는데, student\_list[i]->get\_name 형식끼리 바로 if 문으로 비교하는 형태로 구현하였다가 정렬이 제대로 되지 않는 결과가 발생하였다. 일반적인 이름 변수로는 바로 비교가 안되는 것 같았다. 그래서 string 형으로 먼저 저장해주는 과정을 거치고 그 새로운 변수끼리 비교하는 형식으로 바꾸었더니 정상적으로 정렬이 되는 모습을 확인할 수 있었다.

## Program 3

### □ 문제 설명

string 클래스를 따로 oopstd 란 이름의 네임스페이스로 정의하여 클래스를 구현해보는 문제이다. 생성자는 기본 생성자와 초기화 생성자, 복사 생성자로 총 3 가지로 이루어지며 소멸자에서는 문자열들을 할당 해제시키는 역할을 한다. 그 외 함수는 총 13 개로, 이 중 마지막 4 개는 클래스에는 소속되어있지 않지만 문자열 관련 함수이므로 같이 네임스페이스 안에서 구현하도록 한다.

#### -구현 방법

첫번째로 assign 함수는 문자열을 참조하여, 해당 문자열을 동적 할당하여 현재 값을 저장하고 자신의 객체 값을 반환하도록 만든다. 이때 이미 할당되어 있는 경우에는 먼저 소멸시키고 진행한다. 두번째로 at 함수는 인덱스 값을 받아서 해당 문자열의 인덱스 값을 반환하도록 한다. 세번째로 c\_str 은 string 형에서 char\*형으로 바꿔주는 함수로, 문자열 관련 함수들을 쓰고 싶을 때 많이 쓰이는 함수이다. 함수 반환형 자체가 char\*형이므로 바로 해당 문자열을 반환하면 된다. 네번째로 문자열을 지워주는 함수인 clear 함수는 문자열의 0 번째 인덱스 값을 NULL 값으로 바꾼 뒤 길이 변수도 0 으로 설정해준다. 5 번째로 push\_back 함수는 문자열 끝에 새로운 문자를 추가하는 함수로써, 먼저 길이가 한 칸 더 많은 임시 배열을 생성하고 복사한 뒤, 원래 문자열을 삭제시킨다. 그 다음 임시 배열의 끝 칸에 새로운 문자를 추가한 뒤 마지막 칸을 NULL 값으로 설정하고 다시 기존 문자열에 복사하는 것으로 한다. 6 번째로 문자열 비교 함수인 compare 함수에서는, 먼저 전달받은 문자열의 길이를 따로 저장한 뒤 마찬가지로 임시 배열을 생성하고 복사해준다. const string\*형을 배열로써 조작하기 위해 변환 작업을 거쳐준 것이다. 이후 기존 문자열과 비교를 진행하면서 같으면 0, 기존 문자열이 더 크면 1, 작으면 -1 을 반환하도록 구현한다. 7 번째로 replace 는 문자열 대체 함수로, 똑같이 임시 배열을 생성하여 복사한 다음 길이를 구하고 원하는 값부터 구한 길이 값까지 기존 배열에 임시 배열 값을 넣고 자신의 객체를 반환한다. 8 번째로 substr 에서는 문자열을 추출하는 함수로, 똑같은 과정을 거친 뒤 추출한 배열을 반환해주면 된다. 9 번째로 find 함수에서는 원하는 위치부터 원하는 문자열의 형태가 나올 때까지 탐색하여 인덱스 값을 찾을 수 있도록 구현한다. 이때 문자열이 전부 같아야 하므로, 같을 때 임시 변수를 증가시킨 뒤 나중에 그 변수와 문자열의 길이가

같은지를 따져서 구하는 형식으로 한다. 10 번째는 문자열의 숫자를 정수 타입으로 바꾸는 stoi 함수로, 반복문을 돌면서 숫자 외 문자가 나올 때 그 인덱스 값을 저장하고, 그 값만큼 다시 반복하면서 배열 값에서 아스키코드 '0'값을 뺀 결과에 10 을 곱하고 더하여 숫자로 변환시키도록 한다. 이때 인덱스 값을 매개변수 포인터에 저장하고 합계는 반환하여 같이 쓸 수 있도록 한다. 11 번째의 stof 함수도 문자를 실수로 바꾸는 것이므로, 소수점 위치만 따로 추출한 뒤 이전 이후 부분으로 나누어서 값을 계산하고 반환한다. 12, 13 번째의 정수를 숫자로 변환하는 to\_string 도 위와 같은 내용을 반대로 진행하는 형식으로 구현하도록 한다.

## □ 결과 화면



```
Microsoft Visual Studio 디버그 콘솔
/* assign & c_str */
Hi
/* at */
i
/* push_back */
Hi!
/* replace */
Hi world
/* find */
6
/* stoi */
10
2
/* stof */
1.5
3
/* to_string */
12345
12.345
C:\Users\박나림\OneDrive\바탕 화면\과제\2월3
```

string 클래스의 멤버 함수들과 그 외 함수들 모두 각각 테스트해 본 결과 모습이다.



## □ 고찰

여러 함수들을 구현해보면서 많은 방법들을 시도해봤는데, 일단 제일 문제점이었던 건 전달받은 매개변수의 타입과 그걸 이용하여 구현하고자 할 때 식에 쓰는 타입이 서로 달라 대입이 불가능 하단 점이였다. 그래서 여기서는 형 변환을 이용하거나 주로 임시 배열을 생성하여 먼저 복사해주고 그 배열을 이용하여 계산하는 식으로 진행하였다. 더 좋은 방법이 있을지도 모르니 더 공부해봐야겠다는 생각이 든다. 또한 마지막의 실수를 문자열로 변환하는 방법이 까다로워서 일단 c 언어에 기반한 함수를 이용했는데, 이 부분 역시 추가로 더 고민해봐야 될 것 같다.

## Program 4

### □ 문제 설명

2D 형태의 이차원 배열인 행렬에 대한 연산을 진행할 수 있는 클래스를 구현하는 문제이다. 이때 행렬의 크기에는 제한이 없다. 일반적인 두 행렬에서는 덧셈, 뺄셈, 곱셈 작업을 할 수 있으며 하나의 행렬에서의 스칼라 작업에는 덧셈, 뺄셈, 곱셈, 나눗셈을 할 수 있게 한다. 또한 추가적으로 전치 행렬, 행렬식, 소인수 행렬, 역행렬을 구할 수도 있다. 전 문제와 마찬가지로 네임스페이스에서 따로 행렬 클래스를 구현하는 방식으로 진행한다.

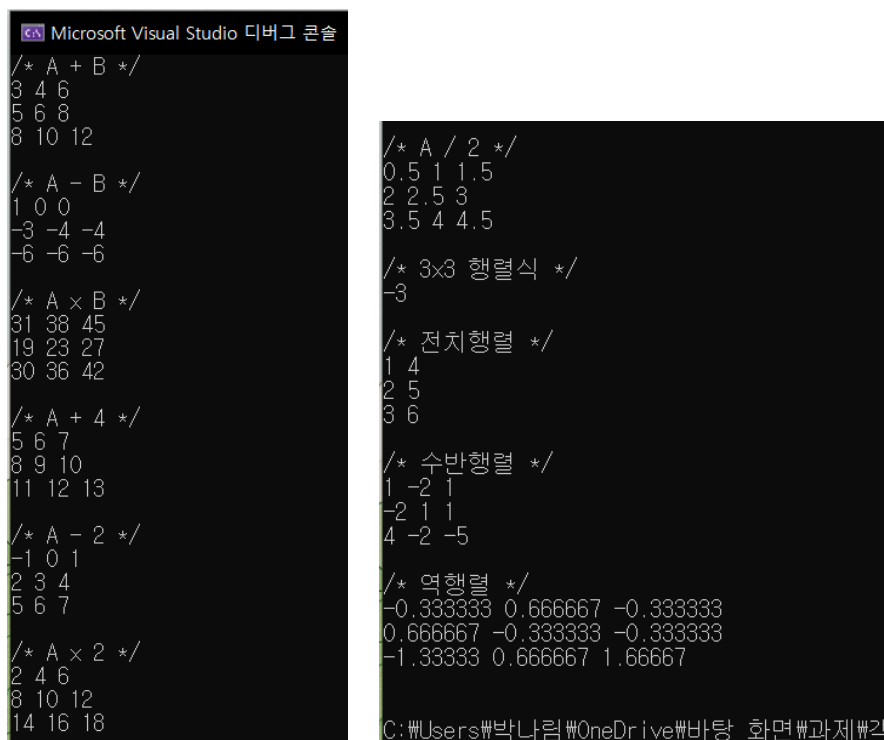
#### -구현 방법

먼저 생성자에서는 멤버 변수들을 다 초기화 시키고, 복사 생성자에서는 매개변수로 받은 행렬의 정보에 따라 배열을 동적 할당하고 깊은 복사를 시켜준다. 이는 나중에 소멸자에서 해제시킬 수 있도록 한다. 각각의 get 함수들에 대해서는 해당 멤버 변수들을 반환할 수 있도록 하며, set 함수들에 대해서는 멤버 변수를 전달받은 매개 변수 값으로 저장해준다. 이때 setData에서는 먼저 원래 가지고 있던 행렬의



동적할당을 해제시켜서 삭제시켜준 다음에 새로운 행, 열 정보로 다시 동적 할당을 시켜주도록 한다. 행렬식 계산 함수에서는 정사각이 아닌 행렬에 대해 예외 처리를 해주고, 행렬의 크기에 따라 케이스를 나누어서 공식을 이용해 진행한다. 그 다음 연산하는 함수들은 각각 매개변수로 받은 행렬의 행과 열을 get 함수로 불러오고, 그를 이용해 예외 처리를 시켜준 다음에 getElement로 배열의 원소 값들을 차례대로 불러오고 연산하여 결과 배열에 setElement 시켜주는 것으로 한다. 그리고 나서 setData로 반환 배열에 저장해 준 뒤 그 배열을 반환해준다.

## □ 결과 화면



```

Microsoft Visual Studio 디버그 콘솔

/* A + B */
3 4 6
5 6 8
8 10 12

/* A - B */
1 0 0
-3 -4 -4
-6 -6 -6

/* A x B */
31 38 45
19 23 27
30 36 42

/* A + 4 */
5 6 7
8 9 10
11 12 13

/* A - 2 */
-1 0 1
2 3 4
5 6 7

/* A x 2 */
2 4 6
8 10 12
14 16 18

/* A / 2 */
0.5 1 1.5
2 2.5 3
3.5 4 4.5

/* 3x3 행렬식 */
-3

/* 전치행렬 */
1 4
2 5
3 6

/* 수반행렬 */
1 -2 1
-2 1 1
4 -2 -5

/* 역행렬 */
-0.333333 0.666667 -0.333333
0.666667 -0.333333 -0.333333
-1.33333 0.666667 1.66667

C:\Users\박나림\OneDrive\바탕 화면\과제\각

```

각 함수들을 구현한 것을 바탕으로 하나씩 실험해 본 모습이다. 행렬은 각각 실습 예제에 나와있는 행렬을 사용하였고, 전치행렬에서는 따로 행렬을 {{1,2,3},{4,5,6}}으로 정의하여 전치시킨 것이다.

## □ 고찰

행렬식을 구하는 함수를 구현할 때,  $3 \times 3$  까지는 공식을 이용해서 진행시켰지만 그보다 큰 행렬에 대해서는 어떻게 해야 할지 잘 모르겠다. 행렬에 대해 다시 공부해보면서 행렬식을 계산할 때 여인수 행렬을 이용하거나 가우스 소거법을 이용하는 등 여러 방법이 있었지만 직접 구현하려니 생각보다 더 어려웠던 것 같다. 비슷한 방법을 쓰는 수반행렬 함수에서는 소행렬을 구할 때 여인수 방법을 이용하는 데에 성공했지만, 행렬식에서는 그 값들을 이용해 곱하고 빼야하는 과정을 거치니, 이 부분에 대해서는 더 고민해봐야 될 것 같다.