



# Object-Oriented Programming Report

Assignment 3-2

Professor	Donggyu Sim
Department	Computer engineering
Student ID	2020202031
Name	Jaehyun Kim

## Program 1

### □ 문제 설명

랜덤한 16 개의 정수를 연결리스트로 저장하고 내림차순으로 정렬한 상태로 이진탐색을 진행하는 프로그램을 작성하는 문제입니다.

존재하지 않는 정수값을 Search 하려고 할 때, 범위 밖의 정수값이라면 양 끝 노드를 반환하고, 범위 내의 정수값이지만 존재하지 않는다면, 그 정수값은 M-1, M 번째 노드에 위치할 것이므로 M-1, M 번째 노드와 정수값의 차를 비교하여 차가 더 적은 노드를 반환하도록 했습니다.

### □ 결과 화면

```
정수 입력: 32 54 24 13 47 86 98 33 15 16 17 97 95 38 75 99
32, 54, 24, 13, 47, 86, 98, 33, 15, 16, 17, 97, 95, 38, 75, 99,
99, 98, 97, 95, 86, 75, 54, 47, 38, 33, 32, 24, 17, 16, 15, 13,
찾고자 하는 정수 입력: 47
47
```

```
정수 입력: 32 54 24 13 47 86 98 33 15 16 17 97 95 38 75 99
32, 54, 24, 13, 47, 86, 98, 33, 15, 16, 17, 97, 95, 38, 75, 99,
99, 98, 97, 95, 86, 75, 54, 47, 38, 33, 32, 24, 17, 16, 15, 13,
찾고자 하는 정수 입력: 99
99
```

```
정수 입력: 32 54 24 13 47 86 98 33 15 16 17 97 95 38 75 99
32, 54, 24, 13, 47, 86, 98, 33, 15, 16, 17, 97, 95, 38, 75, 99,
99, 98, 97, 95, 86, 75, 54, 47, 38, 33, 32, 24, 17, 16, 15, 13,
찾고자 하는 정수 입력: 13
13
```

```
정수 입력: 32 54 24 13 47 86 98 33 15 16 17 97 95 38 75 99
32, 54, 24, 13, 47, 86, 98, 33, 15, 16, 17, 97, 95, 38, 75, 99,
99, 98, 97, 95, 86, 75, 54, 47, 38, 33, 32, 24, 17, 16, 15, 13,
찾고자 하는 정수 입력: 43
Search fail
47
```

```
정수 입력: 32 54 24 13 47 86 98 33 15 16 17 97 95 38 75 99
32, 54, 24, 13, 47, 86, 98, 33, 15, 16, 17, 97, 95, 38, 75, 99,
99, 98, 97, 95, 86, 75, 54, 47, 38, 33, 32, 24, 17, 16, 15, 13,
찾고자 하는 정수 입력: 100
Search fail
99
```

```
정수 입력: 32 54 24 13 47 86 98 33 15 16 17 97 95 38 75 99
32, 54, 24, 13, 47, 86, 98, 33, 15, 16, 17, 97, 95, 38, 75, 99,
99, 98, 97, 95, 86, 75, 54, 47, 38, 33, 32, 24, 17, 16, 15, 13,
찾고자 하는 정수 입력: 5
Search fail
13
```

Search 하려는 정수를 가진 노드를 잘 탐색하여 해당 노드의 데이터 값을 잘 출력하는 것을 볼 수 있습니다. 또한 범위 밖의 정수나, 존재하지 않는 데이터 값을 Search 하려고 시도하면 "Search fail"이라는 문구와 함께 Search 하려는 데이터 값과 가장 근사한 노드를 반환하는 것을 잘 확인할 수 있습니다.

## □ 고찰

배열에서 InsertionSort 는 인덱스값을 가지고 값들을 switch 하지만 연결리스트의 경우 값을 switch 하기 위해서 switch 하려는 노드의 전후 노드의 참조관계를 조작해주고 head 노드도 알맞게 변경시켜줘야 해서 애를 먹었습니다. 존재하지 않는 정수값을 Search 하는 경우에  $L > H$  가 되는 순간  $M$  이 찾으려는 정수 값의 바로 다음 위치에 존재한다는 것을 파악하고 이를 이용하여 코드를 작성했습니다.

## Program 2

### □ 문제 설명

가수, 앨범, 타이틀곡, 발매 년도, 트랙 수 등 음악의 대한 정보를 저장하는 myMusic 클래스를 작성하고 이를 테스트하는 main 코드를 작성하는 문제입니다.

## □ 결과 화면

```
Input by following order.
Singer,Album,Title,Year,Track number.

Input 1st Music: New Jeans,New Jeans,Attention,2022,4
Input 2nd Music: LE SSERAFIM,ANTIFRAGILE,ANTIFRAGILE,2022,5
Input 3rd Music: New Jeans,OMG,OMG,2022,2

1st Music
singer: New Jeans
album: New Jeans
title: Attention
year: 2022
track number: 4

2nd Music
singer: LE SSERAFIM
album: ANTIFRAGILE
title: ANTIFRAGILE
year: 2022
track number: 5

3rd Music
singer: New Jeans
album: OMG
title: OMG
year: 2022
track number: 2
```

,를 구분문자로 하여 가수, 앨범, 타이틀곡, 년도, 트랙 수를 한 줄에 입력하고 이를 ,로 구분하여 각각 myMusic 객체의 멤버변수 m\_singer, m\_album, m\_title, m\_year, m\_track\_no 에 저장하고 차례로 출력하는 것을 확인할 수 있다.

## □ 고찰

year 와 track\_no 을 정수로 입력 받아야 하는데, 문제에서 요구하는 바는 사용자로부터 ,를 구분문자로 하여 한번에 입력하도록 하는 것이었습니다. 그렇다 보니 년도와 트랙 수를 char 형으로 입력 받을 수 밖에 없었습니다. 따라서 문자열로 입력 받은 년도와 트랙 수를 앞 자리수부터 차례로 year 와 track\_no 변수에 저장해줌으로써 년도와 트랙 수를 정수로 저장할 수 있었습니다.

## Program 3

### □ 문제 설명

텍스트 파일에 작성된 artist 를 원형연결리스트로, song 을 연결리스트로 구성하는 문제입니다.

'/'를 경계로 artist 와 song 이 작성돼있는 텍스트 파일에서 한 줄씩 읽어 들인 후

artist 와 song 을 string 형으로 저장한 후 MyMusicManagementList 에서 해당 artist 의 존재 유무를 확인한 후, 이미 존재한다면 존재하는 ArtistNode 에 song 을 추가하고, 존재하지 않는다면 새로운 ArtistNode 를 생성한 후 그 안에 song 을 추가하도록 구현했습니다.

원형 연결 리스트에서 새로운 노드를 추가하기 용이하도록 Head 를 맨 앞 노드가 아닌 맨 뒤 노드로 설정했습니다.

### □ 결과 화면

```

Lim Young Woong      /if we ever meet again
Ed Sheeran           /perfect
NewJeans              /attention      /hype boy
Cho Seong Jin         /debussy: vclaire de lune      /polonaise in a flat major op.53
Mr. Big              /wild world
Tobu                  /higher
Wu-Tang Clan          /tearz
Metallica             /master of puppets (remastered 2016)
Imagine Dragons       /warriors
(G)I-DLE              /latata
Itzhak Perlman        /p. i. tchaikovsky - violine concerto in d major
Ozzy Osbourne         /crazy train
Dream Theater         /pull me under
Pantera               /cowboys from hell
Megadeth              /tornado of souls
Black Pink            /how you like that
Jung Seung Hwan       /fine
Kid Ink               /show me (feat. chris brown)
Big Sean              /blessings (feat. drake)
Rudolf Buchbinder     /beethoven: piano sonata no.8 in c minor op.13 'pathetique' adagio cantabile
Matt Anderson         /ain't no sunshine
Fetty Wap             /679
Wiz Khalifa           /no sleep
G-Eazy                /me, myself & i
TWICE                 /truth or dare
Eminem                /the real slim shady
Rihanna               /diamonds
IU                    /through the night
Bruno Mars            /just the way you are
Maroon 5              /this love
B.O.B                 /nothin' on you
Iron Maiden           /the trooper
Taylor Swift          /speak now
Muse                  /time is running out
Method Man            /what's happenin'
Adele                 /rolling in the deep
Jason Mraz            /i'm yours
50 Cent               /outlaw
Anne-Marie            /2002
Clean Bandit          /rather be
2Pac                  /all eyez on me

```

artist 는 텍스트파일에 입력된 순서대로 원형연결리스트에 나열되고 song 은 소문자로 변환되어 아스키코드 순서대로 연결리스트에 나열된 것을 확인 할 수 있습니다.

Artist / Songs

아무런 Music 정보가 없을 경우 공백만이 출력된다.

Artist	Songs
Lim Young Woong	If We Ever Meet Again
Lim Young Woong	/if we ever meet again

딱 한 곡의 정보만이

존재할 때도 잘 출력된다.

## □ 고찰

원형 연결 리스트를 처음 사용해 보는 것이라, 순회 반복문의 조건을 작성하는데 불편함을 겪었습니다. 하지만 리스트의 Head 를 맨 앞 노드가 아닌 맨 뒤 노드로 설정하는 방식을 알게 되어 이를 활용하여 한결 편하게 조건을 작성할 수 있었습니다. 또한 연결 리스트 내부의 노드에 또 다른 연결 리스트가 존재하는 형태의 코드를 작성하는 것이다 보니, 괜히 복잡하게 느껴지는 부분도 있었지만, 결국은 지금까지 해온 연결 리스트와 다르게 없다는 것을 느꼈습니다.

## Program 4

### □ 문제 설명

다항식의 각 항을 Term 노드에 저장하고 이를 polynomial Linked list 에 저장한 후, 다항식 간 연산을 진행하는 Add, Sub 메소드와 다항식을 출력하는 PrintList 함수를 구현하는 문제다. Add, Sub 메소드는 지수가 같은 Term 끼리 연산을 진행하고, 같은 지수의 Term 이 존재하지 않을 경우에는 기존 polynomial 객체에 연산대상 Term 을 복사하여 Insert 하도록 설계합니다. PrintList 의 경우 계수가 양수이면 계수 앞에 '+' 를 출력하고 음수는 그대로 출력합니다.

### □ 결과 화면

```
p1:  -1x^4 +1x^3 +1x^1 +8 +1x^-1 -9x^-2
p2:  +9x^5 -6x^3 -1x^2 -8x^1 -7 -5x^-2
p3:  -4x^5 +9x^4 -4x^3 +2x^2 +9x^1 -9 -9x^-1 +6x^-2
p1 + p2:  +9x^5 -1x^4 -5x^3 -1x^2 -7x^1 +1 +1x^-1 -14x^-2
p2 - p3:  +13x^5 -9x^4 -2x^3 -3x^2 -17x^1 +2 +9x^-1 -11x^-2
```

같은 exponent 를 가진 Term 끼리 Coefficient 를 연산하여 Add 와 Sub 결과가 올바르게 잘 출력되는 것을 확인 할 수 있습니다. exponent 가 0 이면 상수항이므로 x 를 출력하지 않았습니다.

## □ 고찰

Insert 메소드던, Add, Sub 메소드던, 함수를 구현할 때 제일 처음 노드에 대한 예외사항을 다루는 것이 항상 고민을 불러옵니다. 코드가 모든 상황을 아우르게끔 간결하고 명확하게 작성하고 싶으나, 예외사항을 처리하기 위한 코드가 더해지다 보면 가독성이 점점 떨어질 수 밖에 없기에 어떻게 해야할지 항상 고민이 됩니다. 이번 문제에서는 Add, Sub 메소드를 구현할 때, 연산 결과 계수가 0 이 되는 Term 들을 삭제하는 부분에서 처음 노드에 대한 예외처리를 해주기 위해 예외처리를 해주었습니다.