

# 객체지향프로그래밍

Assignment 3

제출일자: 2018년 05월 20일 (일)

학 과: 컴퓨터공학과

담당교수: 김영민 교수님

학 번: 2014722075

성 명: 이동준

## <1> Singly Linked List

### O Question & Explanation

Make Singly Linked List using class. "LinkedList.h" have class of Node, LinkedList that type is template T. It has function of insertion, insertion in wanted index node, pop, pop in wanted index node, change data in wanted index node and clear. Make function for each function. If the execution of the command does not match the situation, error handling is done

### O Result

Initialization

insertion (1)

insertion (3)

insertion (4)

insertion (6)

Size of linked list = 4

---

idx: 0 , data: 1  
idx: 1 , data: 3  
idx: 2 , data: 4  
idx: 3 , data: 6

---

Test insertion

insertion (1, 2)

insertion (4, 5)

Size of linked list = 6

---

idx: 0 , data: 1  
idx: 1 , data: 2  
idx: 2 , data: 3  
idx: 3 , data: 4  
idx: 4 , data: 5  
idx: 5 , data: 6

---

Test pop

pop (0)

pop (3)

pop ()

Size of linked list = 3

---

idx: 0 , data: 2  
idx: 1 , data: 3  
idx: 2 , data: 4

---

Test get and set function

```
linkedList.get(1)  = 3  
linkedList.set(1, 50)  
linkedList.get(1)  = 50
```

Test clear

```
Size of linked list = 0  
This linked list is empty.
```

계속하려면 아무 키나 누르십시오 . . .

-> program execution based on task data

Initialization

insertion (10)

insertion (100)

Size of linked list = 2

```
idx: 0 , data: 10  
idx: 1 , data: 100
```

Test insertion

insertion (3, 1000)

insertion (0, 1)

Size of linked list = 3

```
idx: 0 , data: 1  
idx: 1 , data: 10  
idx: 2 , data: 100
```

Test pop

pop (10)

pop (1)

Size of linked list = 2

```
idx: 0 , data: 1  
idx: 1 , data: 100
```

Test get and set function

```
linkedList.get(1)  = 100  
linkedList.set(1, 50)  
linkedList.get(1)  = 50
```

Test clear

```
Size of linked list = 0  
This linked list is empty.
```

계속하려면 아무 키나 누르십시오 . . .

-> If error handling is done, not execution of command

## O Consideration

The most popular data structure is a linked list. So we need to know using of linked list using class. To used Template class, make header file of LinkedList.h, fill code using inline method. Based on lecture data, I performed the function correctly for each function. Is was difficult to understand the basic skeleton, but I solved it based on my experience so far.

## <2> Singly Linked List add more function

### O Question & Explanation

Make function reverse and sort using 3-1 assignment. Just addition two function. If linked list exist, execution of reverse function is change direct in linked list. head -> tail ➔ tail(new head) -> head. Execution of sort function is sort node according to data only for ascending order.

### O Result

```
Initialization
insertion (1)
insertion (3)
insertion (4)
insertion (6)
After reverse function
-----
idx: 0 , data: 6
idx: 1 , data: 4
idx: 2 , data: 3
idx: 3 , data: 1
-----
After sort function
-----
idx: 0 , data: 1
idx: 1 , data: 3
idx: 2 , data: 4
idx: 3 , data: 6
-----
Size of linked list = 4
-----
idx: 0 , data: 1
idx: 1 , data: 3
idx: 2 , data: 4
idx: 3 , data: 6
-----
```

-> It is only show reverse function and sort function.

## O Consideration

I have considered the best algorithm to complete the reverse function. I found it. It is partition. I divided it into three parts(only 1 node, only 2 node, 3 and more node). If linked list had 1 node, not need to change. if had 2 node, just change each node. if had 3 and more node, it is need to 3 pointer(o -> o -> o). First change first two nodes, move pointer next 2 node. last change 3 pointer node. Repeat the above steps until the end of linked list. Finally reset head node.

I used Bubble sort of ascending order according to data. Using two for command, I make sort function. It is easy to me, because we already used sort function before assignment.

## <3> Linked list using FILE I/O

### O Question & Explanation

Write program using file and 2D- linked list. File has a list of words(input.txt). And 2D- linked list has alphabetNode and wordNode. alphabetNode is initial of word(ex - a,b,c...). wordNode is words in file. First, make all of alphaberNode, second get words reading file. after get, make word node had word, insert to correct beside alphabernode. wordNode exist from small to large in same initial alphabet.

### O Result

```
a -> accede to -> altercation -> avowal
b
c -> clandestine -> cleavage -> compromise
d -> discord -> divulge -> dovetail
e -> enigma -> estrange -> exploit
f -> fortitude -> friction
g
h
i -> irreconcilable
j
k
l
m
n
o
p
q
r -> reconcile -> relent
s
t
u
v
w
x
y
z
계속하려면 아무 키나 누르십시오 . . .
```

```
divulge
estrange
clandestine
irreconcilable
fortitude
compromise
exploit
reconcile
dovetail
accede to
relent
discord
altercation
enigma
cleavage
avowal
friction
```

-> file had word, it is execution result according to input.txt. file.

#### O Consideration

To solve this question, I used 3 class of linkedList, alphabetNode and wordNode. linkedList class is manager class make linkedList. In linkedList class(function of manager), make all of alphabet node, and make word node after read text file. If read word, find correct alphaber Node, and insert beside alphabet Node. If get all of words in text file, print linkedList it is 2D- linkedList.

i tried to design the program as simple as possible. So used 3 classes. Because there is no skeleton, I have to implement free. But it had the advantage to me. I was able to design easily because I used a variety of linked list algorithm.

#### **<4> Queue**

##### O Question & Explanation

Generate the cards randomly and then put the card in to the Queue. Queue is Linked List. Card has a shape choose one in heart, spade, clover and diamond and number(1 to 13). Each data must be made randomly and differently. Menu is that 1. Generate Card(Push), 2. Delete Card(Pop), 3. Show all card, and 4. End. First set size of Queue by inputting. Next input Menu. Every Menu had error handling about empty or full.

## O Result

Size : 2		
Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End	Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End	
Select menu : 1		
Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End	Select menu : 2 Queue is Empty!!	
Select menu : 1	Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End	
Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End	Select menu : 3	Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End
Select menu : 1 Queue is Full!!!	Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End	Select menu : 2 ◆6 is popped
Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End	Select menu : 1	Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End
Select menu : 2 ♣2 is popped	Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End	Select menu : 3 ♣4
Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End	Select menu : 1	Queue Size : 2 1. Generate Card. 2. Delete Card. 3. Show all card. 4. End
Select menu : 2 ♠12 is popped Queue is Empty	Select menu : 3 ◆6 / ♣4	Select menu : 4 계속하려면 아무 키나

-> can see situation about each menu. it is correctly execution.

can see Queue is Full or Empty.

## O Consideration

To design this program, I used to make random number standard library function. And to make card differently, needed to check already exist or not. Queue is First in – First out, so first in will be inserted last of linked list. it is same push. Pop is delete first in card it is head card. I used variable shape of int. Card's shape is 0~3 to divide and number is 1~13.

This program must have correct exception handling, so each menu has each function. Need to understand about execution of Queue. For making before assignment, I can have knowhow about using linked list. Finally this assignment 3 make me can have confidence to design every linked list algorithm

## <5> MFC, linked list

### O Question & Explanation

Create a "snake game" program using MFC and linked list. When program is started, make snake(size is 3) in the center of map, and map size is 30 by 20. The snake moves according to the direction key. If snake meet wall, game is over. If snake eat "♥", add size of snake.

### O Consideration

When I saw this MFC assignment, I was worried about the completion. I studied based on various data, but I could not complete it. I could not understand the basic method and design method of MFC. I was confident about the linked list, but not on the linked list using MFC.