Object Oriented Programming

Project

Submission date: 2018-06-19

학 과: 컴퓨터공학과

담당교수: 김영민 교수님

실 습: 미수강

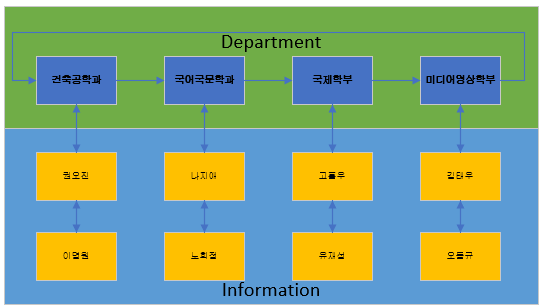
학 번: 2014722075

성 명: 이 동 준

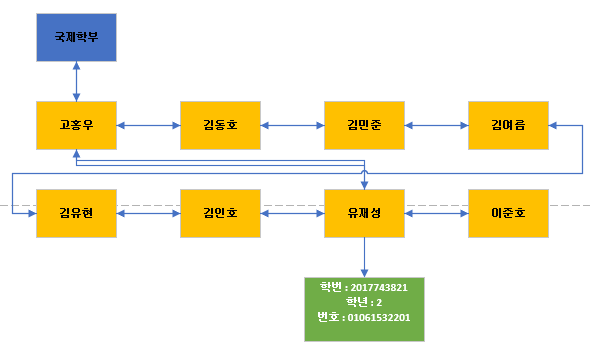
1. Introduction

This program is KW Contact Search System. It is a system that can store the name, student ID and contact information of Kwangwoon University and efficiently search by applying various criteria.

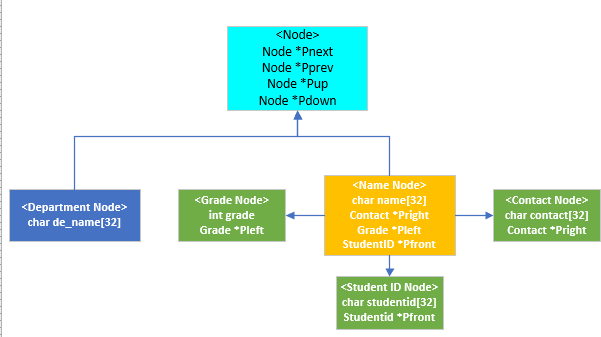
First read command, make linked list. Kinds of Linked list is Circular linked list, Doubly linked list. Department is managed in ascending order using Circular linked list. Under the Department, exist Student had same department using ascending order. Name(student) is managed in ascending order using Doubly linked list. Name have grade, Student ID, contact. They are connected like name. Exist “Node”, it have pointer prev, next, up, down. Department Node and Name Node are inherited classes of the “Node”. But grade, Student ID, contact node is not.



- example of Department and information



- example of Name and student information



- Connection relationship of each node

\* Command

- LOAD : Read “Informatino.txt”, make linked list. File have information of student.

- ADD : Read “Add\_information.txt”, addition information in existed linked list.

- UPDATE : Read “Update\_information.txt”, modify information in existed linked list. It can change only Department, Student Id, Contact.

- MODIFY\_DEPARTMENT : Modify Department name. Can make new Department or combine.

- DELETE\_DEPARTMENT : Delete Department. If delete department, Student had deleted department must delete too.

- PRINT + (default) : Print all of information in existed linked list.

- PRINT\_STUDENT\_ID : Print all of information in existed linked list using insertion sort by student id.

- PRINT\_CONTACT : Print all of information in existed linked list using selection sort by contact.

- FIND\_NAME : Find have inserted name. name can be full name or first name. if inserted first name, find all of student had same first name. after find, print information using bubble sort by name

- SAVE + (textfile) : print in file all of information in existed linked list.

- EXIT : end program and delete memory

\* Error code of Command

- LOAD – 100 : If not exist “Information.txt”.

- ADD – 200 : If not exist “ADD\_information.txt”.

- ADD – 200 -1 : If already exist information.

- UPDATE – 300 : If not exist “Update\_information.txt”.

- UPDATE – 300 -1 : If already exist information.

- UPDATE – 300 -2: If not exist information want update

- MODIFY\_DEPARTMENT – 400 : If not exist department inserted.

- DELETE\_DEPARTMENT – 500 : If not exist department inserted.

- PRINT – 600 -1 : If not exist linked list.

- PRINT – 600 -2 : If not exist department inserted.

- PRINT\_STUDENT\_ID – 700 : If not exist linked list.

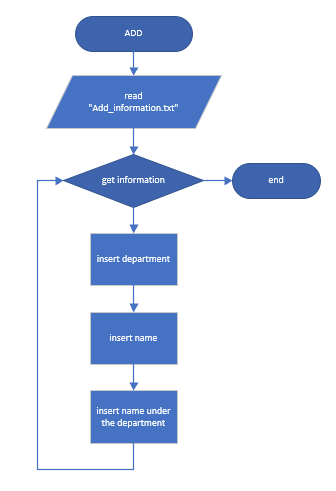
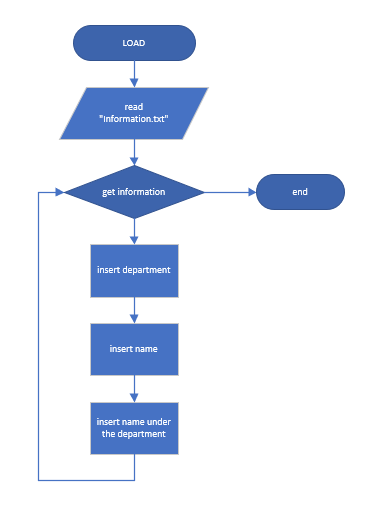
- PRINT\_CONTACT – 800 : If not exist linked list.

- FIND\_NAME – 900 : If not exist full name or first name.

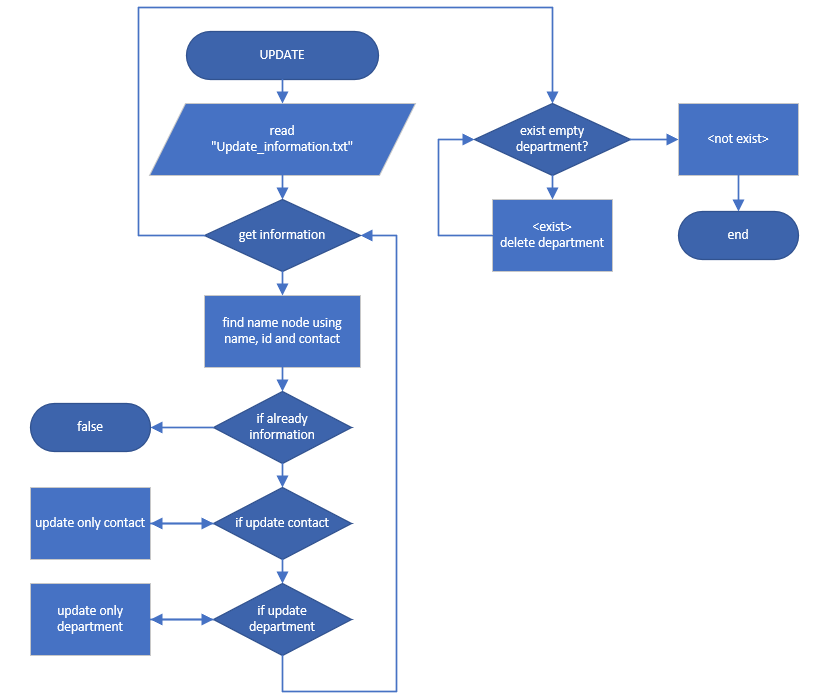
- SAVE – 1000 : If not insert parameter.

- else – 000 : not exist command.

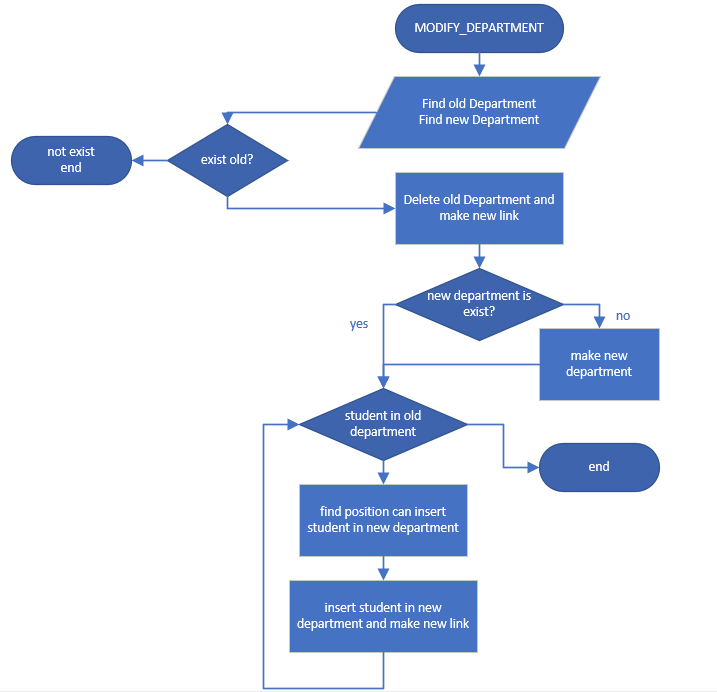
1. Flowchart



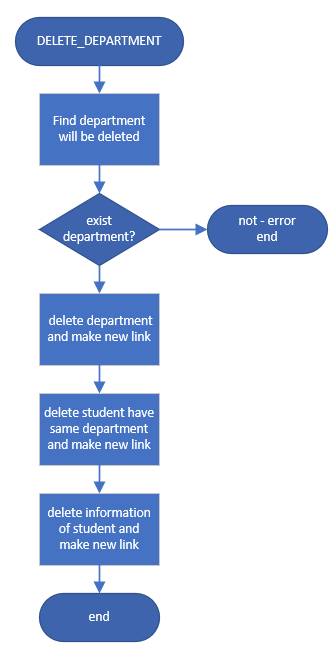
- LOAD function - ADD function



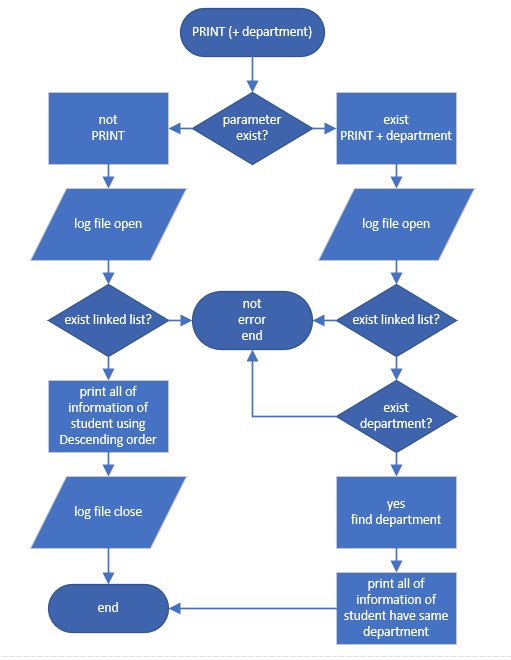
- UPDATE function



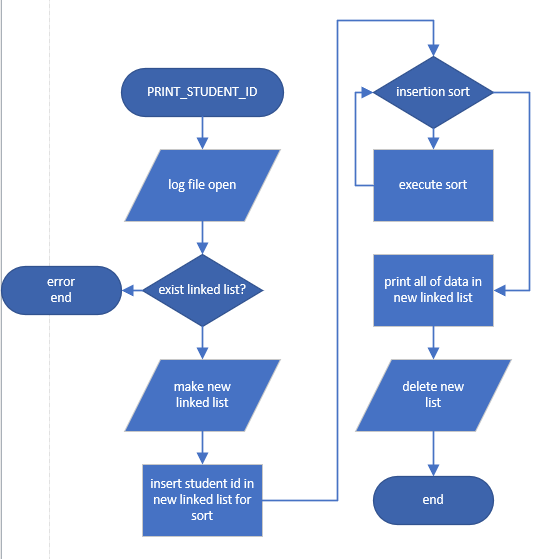
- MODIFY\_DEPARTMENT function



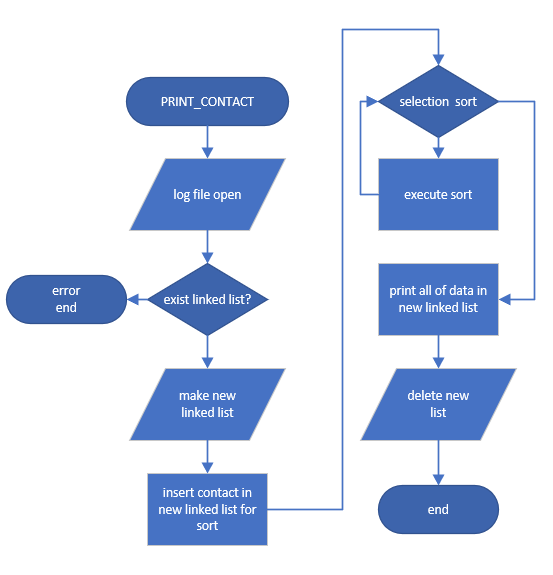
- DELETE\_DEPARTMENT function



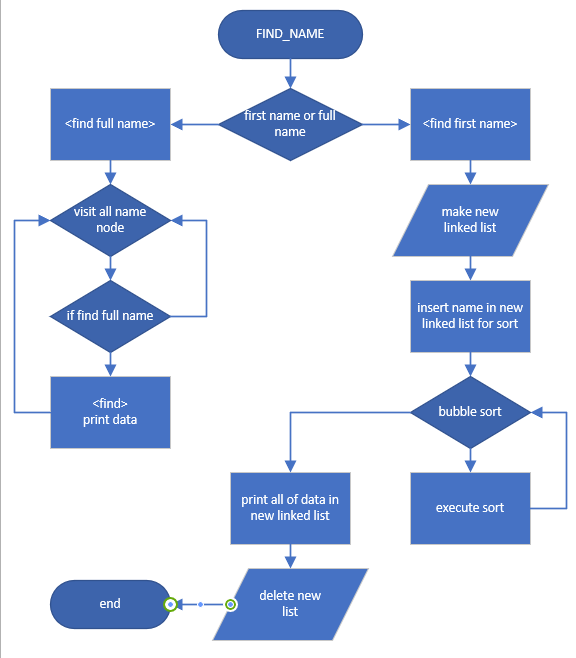
- PRINT (+ DEPARTMENT) function



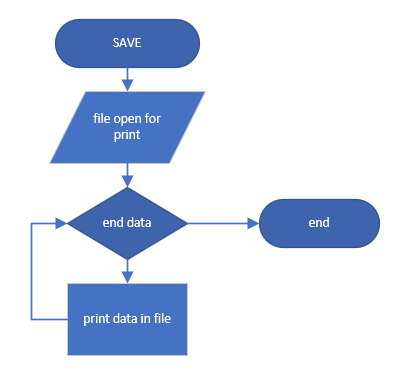
- PRINT\_STUDENT\_ID function



- PRINT\_CONTACT function



- FIND\_NAME function



- SAVE function

1. Algorithm

void DepartmentList::Insert(char \* d)

{

D = find department node using d;

if (D is NULL)

{

make new Departmentnode had name d;

if (linked list is empty)

set head;

else

{

find position can insert;

insert DepartmentNode;

}

}

}

void NameList::insertNameNode(char\* n, char g, char\* s, char\* c)

{

make new namenode had n;

make new gradenode had g;

make new StudentIDnode had s;

make new Contactnode had c;

connect namenode with each information;

if (linked list is empty)

{

set head;

}

else

{

find position can insert name node;

insert namenode and make new link;

make new link each information;

}

}

bool Manager::LOAD()

{

file open("Information.txt");

while (get information)

{

insert DepartmentNode in DepartmentLinkedlist;

insert NameNode in NameLinkedlist;

find DepartmentNode in DepartmentLinkedlist;

find NameNode in NameLinkedlist;

connect DepartmentNode with NameNode;

}

}

int Manager::ADD()

{

file open("Add\_information.txt");

while (get information)

{

insert DepartmentNode in DepartmentLinkedlist;

insert NameNode in NameLinkedlist;

find DepartmentNode in DepartmentLinkedlist;

find NameNode in NameLinkedlist;

connect DepartmentNode with NameNode;

}

}

int Manager::UPDATE()

{

file open("Update\_information.txt");

while (get information)

{

find namenode;

if (already exist information)

error;

else

{

find Department;

update each information;

}

}

}

bool Manager::MODIFY\_DEPARTMENT(char\* oldD, char \* newD)

{

find old Departmentnode;

find new Departmentnode;

if (old is not exist)

error;

if (new is not exist)

{

modify old name to new;

move Departmentnode in new position;

}

else

{

delete old Department;

insert namenode under the new Department;

}

}

bool Manager::DELETE\_DEPARTMENT(char \* delD)

{

find Department wiil be deleted;

if (not exist)

error;

make new link;

delete Namenode under the Department will be delete;

delete Department;

}

void Manager::PRINT()

{

file open("Log.txt");

while (all of department)

{

while (namenode under the department)

{

print information;

}

}

}

void Manager::PRINT(char\* Dname)

{

file open("Log.txt");

find Department;

if (not exist)

error;

while (namenode under the Department)

{

print information;

}

}

void Manager::PRINT\_STUDENT\_ID()

{

file open("Log.txt");

if (not exist linked list)

error;

make new linked list using Namelist;

insert node had data of StudentID;

insertion sort by student id;

print new linked list

}

void Manager::PRINT\_CONTACT()

{

file open("Log.txt");

if (not exist linked list)

error;

make new linked list using Namelist;

insert node had data of Contact;

selection sort by student id;

print new linked list

}

bool Manager::FIND\_NAME(char \* name)

{

file open("Log.txt");

if (find full name)

{

while (visit all of namenode)

{

if (find)

print information;

}

}

else //find first name

{

make new linked list;

while (visit all of namenode)

{

if (find name had same first name)

{

insert node in new linked list;

}

}

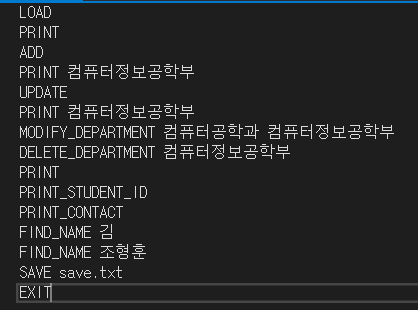
bubble sort by full name;

print all of data in new linked list;

}

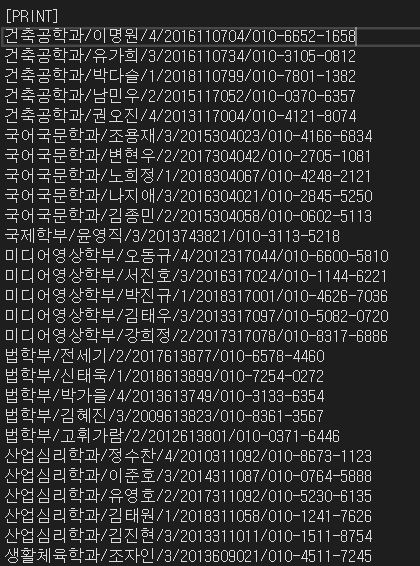
}

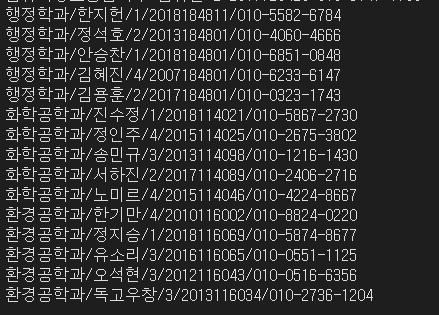
1. Result Screen

 -> Command

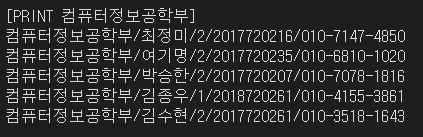
<In Log.txt>

 -> result of LOAD

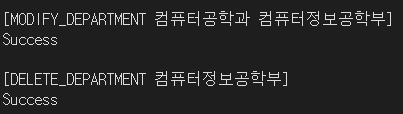


 -> result of PRINT

 -> result of ADD

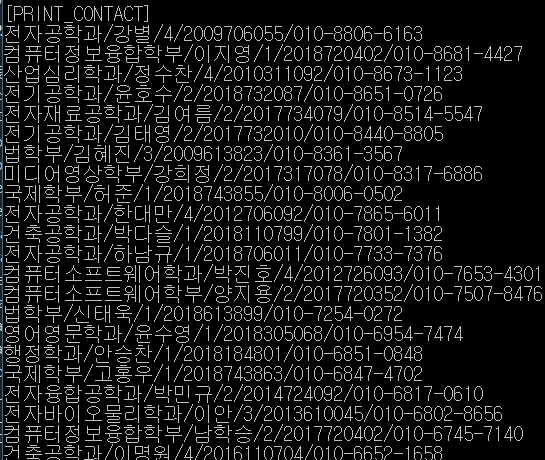
 -> result of PRINT + department

 -> result of UPDATE, it had wrong information

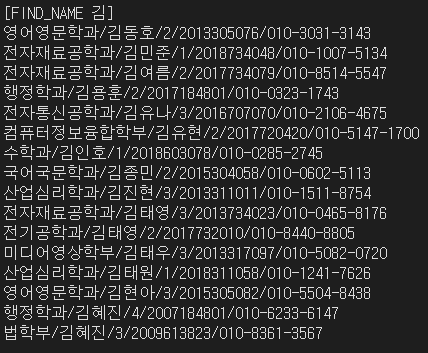


-> result of MODIFY and DELETE DEPARTMENT

-> result of PRINT\_STUDENT\_ID

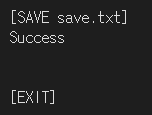


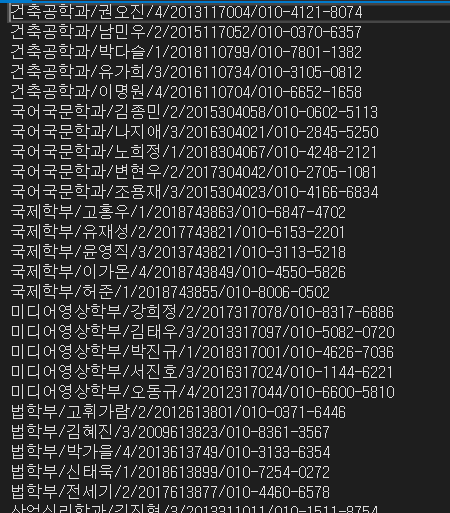
-> result of PRINT\_CONTACT



-> result of FIND\_NAME + first name

 -> result of FIND\_NAME + full name

 -> result of SAVE and EXIT

 -> save.txt

1. Consideration

To solve this project, I need to know using various linked list. So I studied Double, Circuit, 2D linked list. When I see this project, I thought it is similar with data structure project. I attended data structure, I can using code when I made. This project has complex linked list. Very important thing is that in this project. In addition, linked list sort method is important. I can finish using lecture materials of OOP. I focused on the simplification of the code. So, the necessary variables are set individually and I made it possible to classify the various situations and execute the appropriate action for each situation. The division of the situation is very important because there are various and complex linked list. After the design is finished, all memory was deleted through the destructor.

Through this object-oriented design, the theoretical part is more accurate than the design method. I should be able to take advantage of the C++ features, the ability to design high-efficiency algorithms should be developed. I must maintain and elevate this ability and make it more capable than others.