Object Oriented Programming

Project

Submission date: 2018-06-17

학 과: 컴퓨터공학과

담당교수: 김영민 교수님

실 습: 수강/목요일

학 번: 2015722025

성 명: 정 용 훈

1. Introduction

**(1)**

The main content of this project is the linked list. In the course of creating the project topic of address book, the department used the singly linked list to link the address books, and in order to create lists of names, we used the double-linked list. In addition, each student's information - grade, class, and phone number - is created separately, and the link between the information is connected to the singly-linked list to complete the entire address book. In particular, the single-linked list, which connects the department, has a circular list that connects the head and tail. The above is the whole story of the project.

**(2)**

Next, as you configure your project, you receive a command, which is a description of each function.

**LOAD:** As mentioned earlier in (1) It’s a function that makes up the information. In particular, it is a function to import stored information into a program using IO and organize the list accordingly. In particular, the names of departments and departments are associated in ascending order as a condition for the list to be configured.

**ADD:** This function is executed when the command ADD is received after configuring the Load described earlier. The file Addinformation.txt is configured separately and provides the ability to add new information to existing lists for students that you want. Some of the existing information should not be overlapped by a student number or cell phone number.

**UPDATE:** Commands that change student information by name. This function allows the existing information to be overwritten with new information. The exception process to be noted is that if a course is updated and there are no students in the department, it must be eliminated

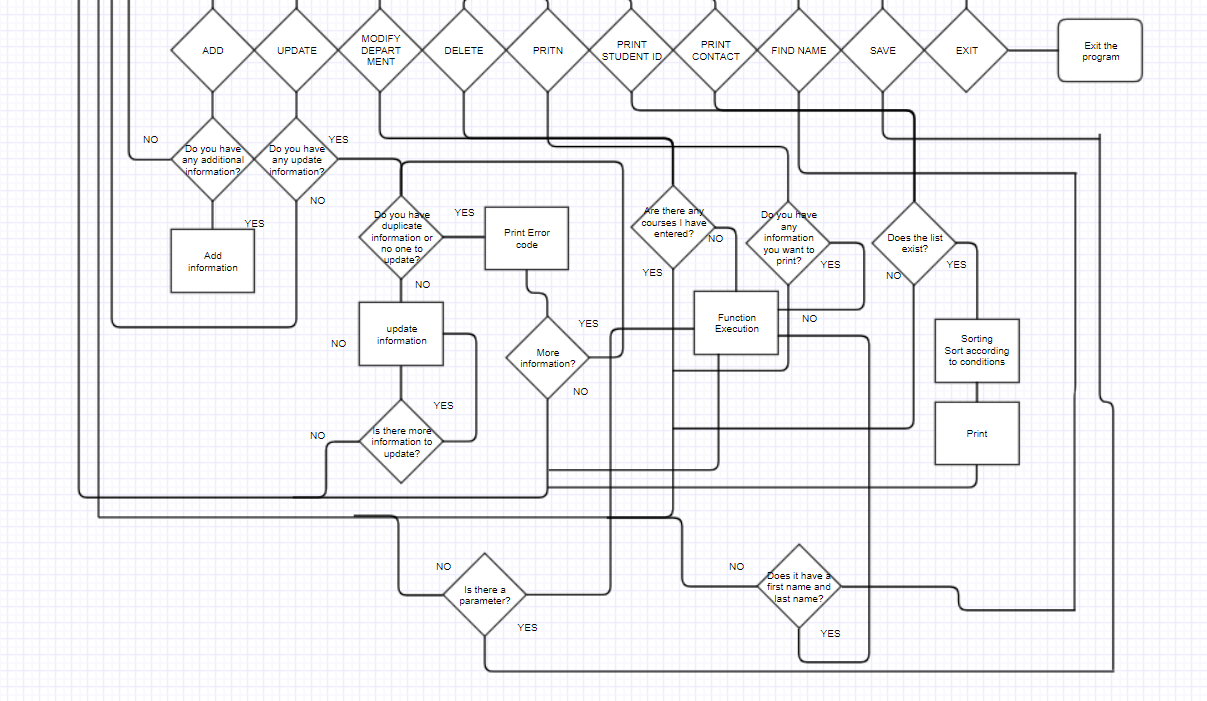
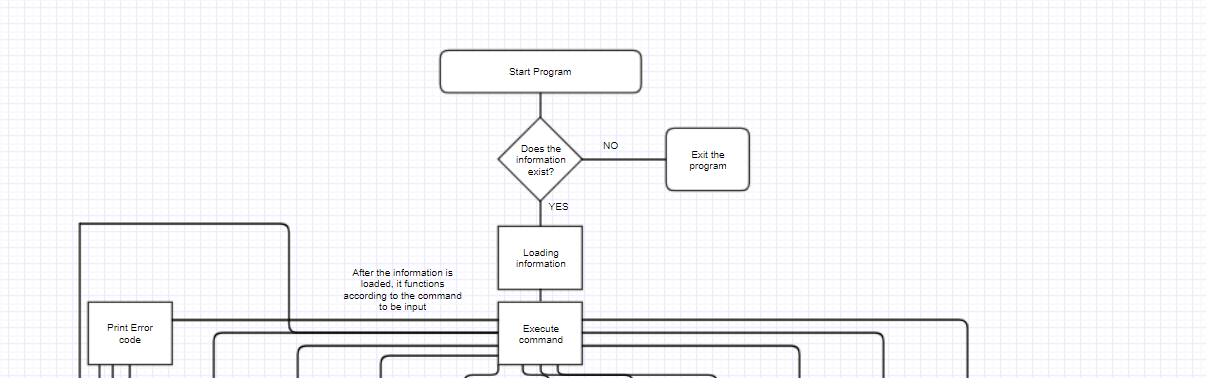
**MODIFY:** As a parameter, you will receive two subjects. This command changes the preceding subject to the next one. If the name is not already changed, simply rename the department, but if the subject is original, it will be necessary to bring students together and to convert them in ascending order.

**DELETE:** You will receive a course as a parameter and remove it. It is important to note that students in the department should also be deleted and that you should pay attention to data leakage.

**SAVE:** A command that performs commands and saves the completed address book as a txt file.

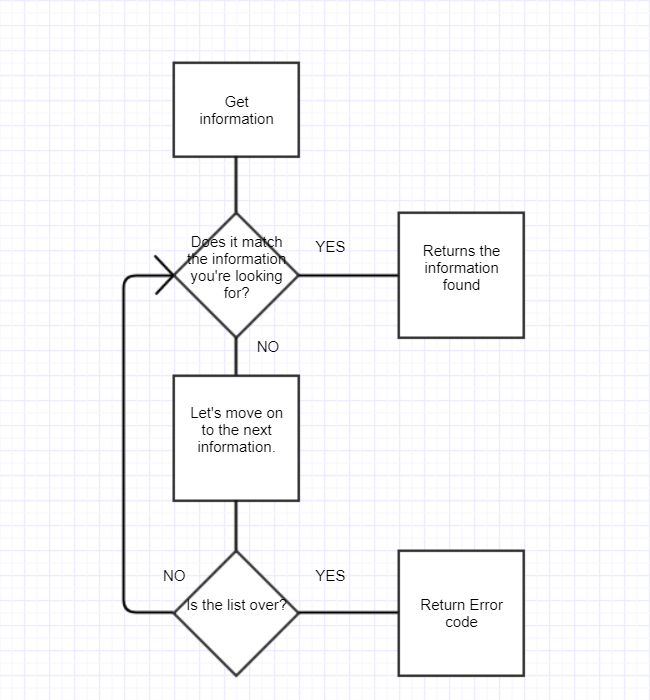
**EXIT:** You will end the program when you receive the command. It looks like a simple function, but it is the most important command because you need to delete all your data.

1. Flowchart
2. Flow chart for command



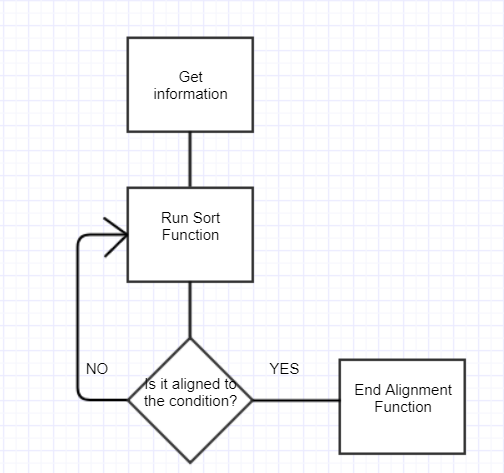
1. Flow chart used in most functions

The flowchart below is a method used to find information.



1. A flow chart used for alignment

This is a function used to sort departments and names.



1. Algorithm

The algorithms used in this project are as follows:

1. Algorithm for commands

A feature of this project is to receive and execute commands as a txt file. So, the algorithm was implemented by repeating the reading of the command until the EXIT function came out, and then executing the corresponding function if the command was matched.

1. Algorithm for locating incoming information

In many of these assignments, we found the same information and then modified it. So the algorithms for finding information were especially used. As a method, we solved the problem by comparing the input information from the head of the list sequentially using the compression function until the 0 is returned.

1. Algorithm for making lists of department and names

As a condition for creating lists of departments and names, the list had to be linked in ascending alphabetical order. So, I started to connect all the information in sequence and then compare the information in the department with the name, and then I created a function of sorting. If you allow the two preceding nodes to be compared sequentially until all lists are aligned, they will eventually be sorted in ascending order. The process was repeated to meet the conditions.

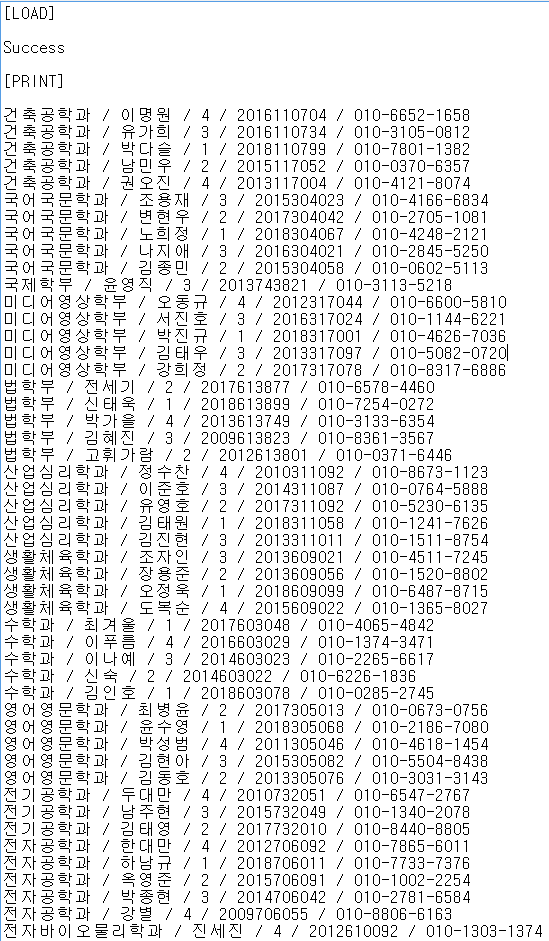
1. Insertion alignment algorithm

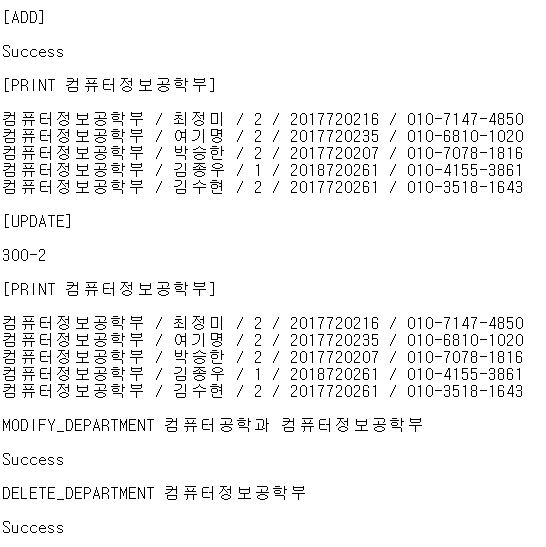
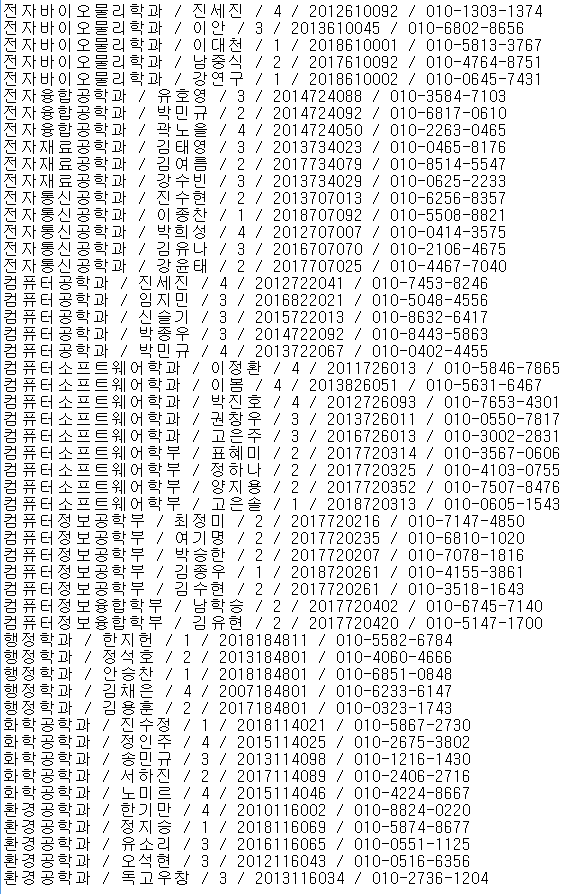
In this task, I created a function by creating a new list, printing it, and reselecting it in order. So, when you do the insertion sort, you pass on the information as a function that allows you to create the list sequentially from the original list. The list creation function has created an algorithm that allows you to determine what to put before or after compared to the node you had before.

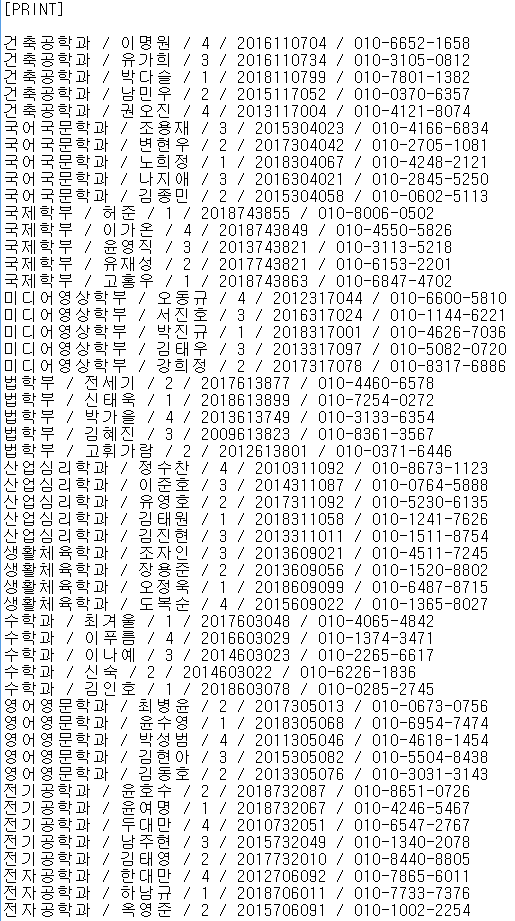
1. Selection sorting algorithm

Selection sorting is the opposite of the insertion alignment. We gave the information with the maximum value to the function that made the list first, then we found the information that was the maximum, and then we sent it to the function that made the list, and we used the algorithm to find the maximum.

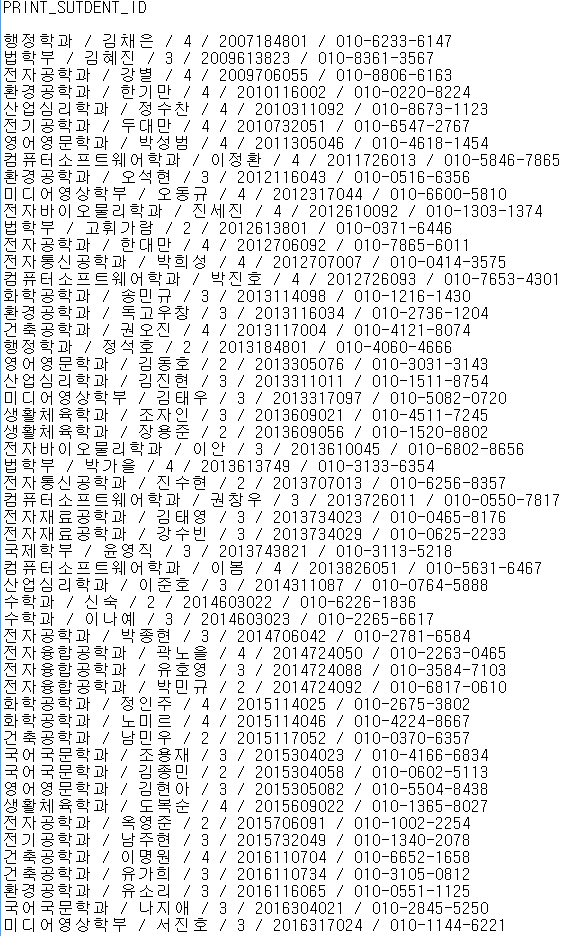
1. Result Screen



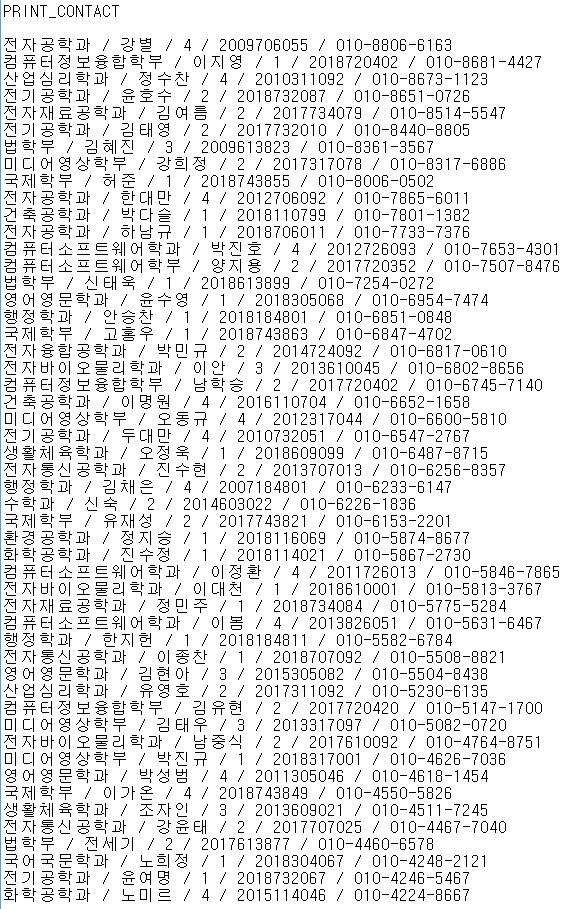


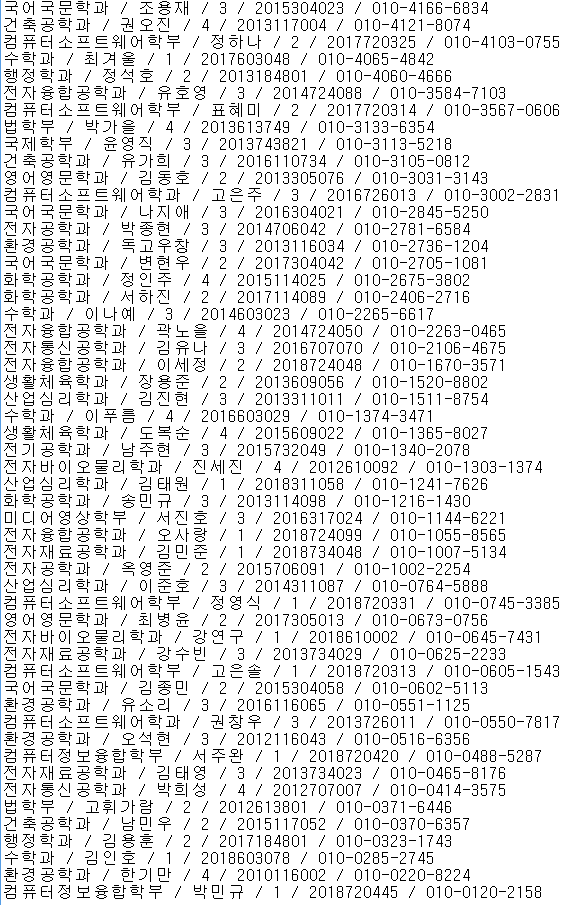


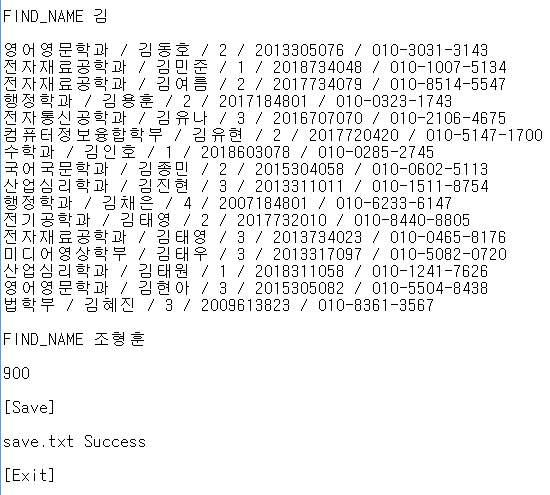












1. Consideration

The main topic of this project is the linked list. A program that receives the information, creates the list according to the appropriate conditions, and runs the commands sequentially according to the command in the txt file to modify the list, save, or exit the program. In particular, in this task, we used the inheritance of class to link the list, but at first, we were not sure how to approach other data types, so it was very difficult to link the list. The issue was resolved by declaring Get and Set functions virtual and the connection was simplified. Secondly, it was done in a slightly unusual way in the alignment of inserts and selection. The method was to obtain the corresponding information from the original list and create a new list according to the appropriate conditions, and then erase the list again. The caution is to create a new list through dynamic allocation, so be careful to release memory. Finally, leaks to memory were also very troublesome. After completing the program, we checked, and it took us a long time to find where the leak was coming from. The next time you create a program, I think using a pre-checking function to prevent leakage of memory is a more obvious and accurate method.