2023 Spring OOP Assignment Report

과제 번호 : #2
학번 : 20220923
이름 : 이윤혁
Povis ID : leeyoonhyuk0

명예서약 (Honor Code)

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

1. 프로그램 개요

이 프로그램은 학생 데이터를 다룬다. 학생 데이터를 입력하거나 삭제할 수 있으며, 학생 데이터를 이용해 목록을 출력하거나 피벗 변환을 수행해 통계를 낼 수 있다.

프로그램을 실행하면 메뉴 리스트가 출력되고, 메뉴의 숫자를 입력해 메뉴를 선택할 수 있다. 1. Add a student를 선택하면 학생을 추가할 수 있고, 학생의 Department, Gender, Name, Age를 입력할 수 있다. 2. Delete a student를 선택하면 학생을 제거할 수 있고, 학생의 Department, Gender, Name, Age를 입력할 수 있다. 3. Print the student's list를 선택하면 입력된 학생들을 Departmet, Gender, Name, Age를 포함해 리스트로 출력한다. 4. Pivot Table를 선택하면 피벗 변환할 기준과 피벗 변환으로 얻을 통계 값을 선택할 수 있다. 선택하면 입력된 기준과 통계 값으로 피벗 변환을 수행해 기준과 통계값을 리스트로 출력해준다. 5. Exit을 선택하면 프로그램이 종료된다.

본 프로그램의 파일은 [표 1]에서 설명한다.

[표 1] 프로그램 구성파일 설명

파일명	설명
list.cpp	list class를 정의하는 파일
list.hpp	list class를 선언하는 파일
main.cpp	메인 프로그램 파일
node.cpp	node class를 정의하는 파일
node.hpp	node class를 선언하는 파일
student.cpp	student class를 정의하는 파일
student.hpp	student classs를 선언하는 파일

2. 프로그램의 구조 및 알고리즘

2.1. Class

본 프로그램에서 사용하는 클래스에 대해서 설명하겠다.

2.1.1. student

student는 학생의 데이터를 저장하는 클래스이다. [표 2]는 student의 멤버 변수를 설명한다. [표 3]는 student의 멤버 함수를 설명한다.

[표 2] student의 멤버 변수

접근제한자 자료형 변수명	설명
private string dept	학생의 학과
private string name	학생의 이름
private string gender	학생의 성별
private Int age	학생의 나이

[표 3] student의 멤버 함수

접근제한자 반환자료형 함수명(파라미터)	설명
public student()	student의 default constructer
public void input_info(string, string, string, int)	학생의 정보를 입력받는 함수
public string get_dept()	dept의 accessor
public string get_gender()	gender의 accessor
public string get_name()	name의 accessor
public Int get_age()	age의 accessor

2.1.2. node

node는 student를 보관하는 연결 리스트의 노드 클래스이다. [표 4]는 node의 멤버 변수를 설명한다. [표 5]는 node의 멤버 함수를 설명한다.

[표 4] node의 멤버 변수

접근제한자 자료형 변수명	설명
private student data	학생 데이터 클래스
private node* next	다음 노드의 포인터

[표 5] node의 멤버 함수

접근제한자 반환자료형 함수명(파라미터)	설명
public node()	node의 default constructer
public node(string, string, string, int)	node constructer, student의 정보를 입력할 수 있다.
public void set_next(node*)	next의 mutator
public student get_data()	data의 accessor
public node* get_next()	next의 accessor

2.1.3. list

list는 node를 노드를 가지는 연결 리스트 클래스이다. [표 6]는 list의 멤버 변수를 설명한다. [표 7]는 list의 멤버 함수를 설명한다.

[표 6] list의 멤버 변수

접근제한자 자료형 변수명	설명
private int count	연결 리스트의 노드의 개수
private int dept_cnt	연결 리스트에 있는 학생의 학과의 개수
private string dept[9]	연결 리스트에 있는 학생의 학과 배열
private int dept_member_num[9]	연결 리스트에 있는 학과 별 학생수 배열
private node* head	연결 리스트의 head

[표 7] list의 멤버 함수

접근제한자 반환자료형 함수명(파라미터)	설명
public list()	list의 default constructer
public ~list()	list의 default deconstructer, 리스트가 삭제될 때 모든 노드를 할당해제한다.
public bool is_dept_full()	연결 리스트의 학과가 다 찼는지 판별하는 함수
public bool is_dept_exist(string)	학과가 연결 리스트에 있는지 판별하는 함수
public bool is_dept_member_full(string)	학과의 학생이 다 찼는지 판별하는 함수
public bool is_student_exist(string, string, string, int)	학생이 연결 리스트에 존재하는지 판별하는 함수
public void save_node(node*)	연결 리스트에 노드를 저장하는 함수
public void delete_node(node*)	연결 리스트에 있는 노드를 삭제하는 함수
public node* find_node(string, string, string, int)	학생 정보를 통해 연결 리스트에 있는 노드를 탐색하는 함수

public void sort(string)	주어진 기준으로 연결 리스트를 정렬하는 함수
public node* get_head()	head의 accessor
private node* find_parent(node*)	주어진 노드의 parent 노드를 탐색하는 함수
private node* merge_sort(node*, int, string)	연결 리스트를 병합 정렬하는 함수
private node* merge(node*, node*, string)	병합 정렬 중 분할 된 연결 리스트를 병합하는 함수
private node* pop_first()	연결 리스트의 첫 번째 노드를 pop해오는 함수

2.2. main

main()은 main.cpp에 선언 및 정의되어 있다. [표 8]은 main에서 사용하는 변수를 설명한다. [표 9]는 main.cpp에 정의 된 함수를 설명한다.

[표 8] main()의 변수

자료형 변수명	설명
const int ADD_STUDENT	상수
const int DELETE_STUDENT	상수
const int PRINT_STUDENT_LIST	상수
const int PIVOT_TABLE	상수
const int EXIT	상수
const int DEPT	상수
const int GENDER	상수
const int DEPT_AND_GENDER	상수
const int AVERAGE	상수
const int MAX	상수
const int MIN	상수
int selection	선택한 메뉴를 저장하는 변수
int category_selection	피벗 변환에서 선택한 카테고리를 저장하는 변수
int function_selection	피벗 변환에서 선택한 함수를 저장하는 변수
bool is_exception_throwed	예외처리가 일어났는지 판별하는 변수
string dept	입력 받은 학과
string gender	입력 받은 성별
string name	입력 받은 이름
int age	입력 받은 나이
node* temp	임시 노드

list student_list	학생 정보을 담는 연결리스트

[표 9] main()의 함수

반환자료형 함수명(파라미터)	설명
void print_menu()	메뉴 리스트를 출력하는 함수
void print_category()	피벗 변환 카테고리 리스트를 출력하는 함수
void print_function()	피벗 변환 함수 리스트를 출력하는 함수
void print_dept_pivot_table(list*, int)	학과 피벗 변환 결과를 출력하는 함수
void print_gender_pivot_table(list*, int)	성별 피벗 변환 결과를 출력하는 함수
void print_dept_and_gender_pivot_table(list*, int)	학과와 성별 피벗 변환 결과를 출력하는 함수
bool is_selection_in_range(int)	메뉴 선택이 올바른지 판별하는 함수
bool is_sub_selection_in_range(int)	카테고리, 함수 선택이 올바른지 판별하는 함수
bool is_all_uppercase(string)	입력한 문자열이 모두 대문자인지 판별하는 함수
bool is_all_alphabet(string)	입력한 문자열이 모두 영어인지 판별하는 함수
bool is_gender_valid(string)	입력한 성별이 올바른지 판별하는 함수
bool is_age_in_range(int)	입력한 나이가 올바른지 판별하는 함수

입력의 예외처리는 [표 10]과 같이 수행했다.

[표 10] 에외처리 수행

예외상황	대응방안
selection이 정수가 아님	재입력 요청
1 <= selction <= 5	재입력 요청
age가 정수가 아님	재입력 요청
학과가 이미 9개임	학생입력 거부
학과 구성원이 이미 10000명임	학생입력 거부
학과 이름이 대문자가 아님	학생입력/학생삭제 거부
학과 이름이 영어가 아님	학생입력/학생삭제 거부
성별이 잘못됨	학생입력/학생삭제 거부
학생 이름이 영어가 아님	학생입력/학생삭제 거부
나이가 주어진 범위를 벗어남	학생입력/학생삭제 거부
입력한 학생이 이미 존재함	학생입력 거부
입력한 학생이 존재하지 않음	학생삭제 거부

Main()은 선택한 메뉴를 수행한다. 각 메뉴마다 어떻게 작동하는지 설명하겠다.

2.2.1. Add a student

입력할 학생 정보(학과, 성별, 이름, 나이)를 순서대로 입력받는다.

학생 정보가 예외처리되지 않은 경우 노드를 동적할당하고, 노드에 학생 정보를 입력한다.

2.2.2. Delete a student

삭제할 학생 정보(학과, 성별, 이름, 나이)를 순서대로 입력받는다.

학생 정보가 예외처리되지 않은 경우 일치하는 학생 노드를 탐색하고, 할당해제한다.

2.2.3. Print the student's list

연결리스트를 학과, 성별, 이름, 나이 순으로 정렬한다. 정렬에는 병합 정렬을 사용한다.

색인표를 출력하고, 노드 안에 있는 학생 정보를 순서대로 출력한다.

2.2.4. Pivot Table

피벗 변환을 수행하기 위해 카테고리 및 함수를 선택한다.

입력된 카테고리를 기준으로 연결 리스트를 정렬한다. 정렬에는 병합 정렬을 사용한다.

색인표를 출력하고, 선택한 함수를 이용한 피벗 변환 값을 순서대로 출력한다.

2.2.5. Exit

프로그램을 종료한다. 프로그램 종료시 리스트에 있는 모든 노드를 할당해제한다.

3. 토론 및 개선

이번 과제를 해결하면서 캡슐화 및 정보 은닉을 직접 구현해볼 수 있었다. 또한 CLI 프로그램을 만들면서 사용자와 상호작용하는 방법과 잘못된 입력의 예외처리가 중요함을 알 수 있었다.

연결 리스트의 학생정보를 txt 파일이나 csv 파일로 저장하는 기능을 추가하면 프로그램을 개선할 수 있을 것 같다.

연결 리스트를 정렬할 때 병합 정렬을 사용하였는데, 적은 노드 수에서는 선택정렬이 더 빠르기 때문에 특정 노드 수(15)를 기준으로 삼아 노드 수가 적으면 선택정렬, 노드 수가 많으면 병합정렬을 수행하면 정렬이 더 빠르게 수행할 수 있을 것 같다.

4. 참고 문헌

Stephen Prata. (2011). C plus plus Primer Plus (6th edition)

2022F CSED233 Data Structure lecture slides

2023S CSED232 Object Oriented Programming lecture slides