실습 과제 2번이 쉽지 않지요?

과제를 시작한 분들은 지금쯤은 많은 고민에 쌓여 있을 것 같습니다.

여러분들의 고민을 해결하는데 도움을 주기 위해서 김병우 조교가 예제를 만들어 봤어요. 그렇다고 이 예제가 정답이라고 볼 수 없고 설사 정답이라고 해도 소스코드를 모두 공개할 수는 없어요.

고민을 해결에 도움이 될 수 있는 몇가지 내용을 요약했으니 참조하세요.

* Folder class가 folder 리스트를 멤버로 갖게 되면 메모리가 반복적으로 할당되어 오류가 발생할 수 있습니다. 이 문제를 해결하기 위해서는 다음 방법을 생각해 볼 수 있습니다.
  1. Folder class의 folder list를 sorted list pointer로 선언하는 방법
     + 폴더가 추가될 때 객체를 동적으로 할당한다.
  2. Sorted list에 멤버인 ItemType 배열을 포인터 배열로 선언하고 item에 추가될 때 item을 저장할 메모를 할당하는 방법
  3. 기타 여러 방법이 있음
* 위 방법 중에서 1번 방법을 구현하기 위한 힌트를 아래에 제시함
* Application class와 folder class의 설계:
  + 방법 1: Application class에서는 root 폴더를 생성하고 바로 root folder로 제어를 넘김
    - 각 폴더가 사용자 인터페이스를 가지고 해당 폴더를 제어하고
    - Application class는 폴더 전체를 관리하고 검색하는 기능을 부여
  + 방법 2: 사용자와 interaction 하는 기능을 Application class에 부여
    - 폴더 class는 application class를 지원하는 기능 수행하는
* 방법 1에 대한 힌트를 아래에 제시함

**Folder class의 folder list를 sorted list pointer로 선언하는 방법**

1. Folder Class

class FolderType

{

public:

FolderType() { fdSubFolderNum = 0; 기타 멤버 초기회 }

~FolderType() { fdFolderList 해제 }

void SetRecordFromKB(); // 키보드로부터 전체 정보를 입력

RelationType CompareByID(const FolderType& data);

FolderType& operator= (const FolderType& data);

int AddFolder(); // 새폴더 정보를 키보드로부터 읽어서 폴더리스트에 추가,

//처음 폴더이며 sorteList를 동적으로 할당하여 초기화 하고 추가

int DeleteFolder(); // 삭제할 폴더 정보를 읽어서 해당 폴더를 찾아서 삭제

int RetrieveFolderByName();// 검색할 폴더정보를 화면에서 읽어 검색하여 결과 출력

void DisplayAllFolderName();// 폴더정보 화면출력

// 폴더 멤벼변수 일출력 함수들 정의

private:

int fdSubFolderNum;

string fdPath;

string fdName;

string fdCreatTime;

SortedArrayList\* fdFolderList;

//기타 필요한 폴더 정보 정의 ;

};

AddFolder() 함수 예

int FolderType::AddFolder()

{

if (fdSubFolderNum == 0)

fdFolderList = new SortedArrayList;

FolderType data;

data.SetPath(fdPath+”/”+fdName); // 서브폴더 패스 설정

data.GenCreateTime();

//data folder 기탄 변수를 키보드로부터 읽어 들인다.

if (fdFolderList->Add(data))

fdSubFolderNum++;

return 1;

}

// generate the name of a photo file based on the current time.

// It generates 14 digit file name yyyymmddhhmmss

void int FolderType::GenCreateTime()

{

// Declaring argument for time()

time\_t tt;

// Declaring variable to store return value of

// localtime()

struct tm \* ti;

// Applying time()

time(&tt);

// Using localtime()

ti = localtime(&tt);

// stream output

stringstream ss;

ss << setw(4) << ti->tm\_year + 1900 << setfill('0') << setw(2)

<< ti->tm\_mon + 1 << setfill('0') << setw(2) << ti->tm\_mday << setfill('0') << setw(2) << ti->tm\_hour << setfill('0') << setw(2)

<< ti->tm\_min << setfill('0') << setw(2) << ti->tm\_sec << "\0";

fdCreatTime = ss.str(); // copy the stream buffer to name

}

Application class와 folder class의 구분 힌트

* Application class
  + Root folder를 생성하고 초기화 한다.
  + main에서 root folder를 application class를 정의하고 제어를 root folder의 Run함수로 넘긴다.
  + 나중에 전체 폴더에 적용되는 기능이 필요하면 application의 Run함수에 추가하여 호출한다.
* Folder class
  + 서브 폴더의 생성, 삭제, 갱신, 검색 함수를 멤버 함수로 갖는다.
  + Run 함수에서 사용자와 interaction을 한다.