**자료구조**

**프로젝트 2 명세서**

컴퓨터공학과

2015104147 공재호

# 프로젝트 주제

사진, 음원, 동영상을 관리하기 위한 폴더 시스템을 구현하는 것이다. 폴더안에 폴더와 파일이 존재할 수 있다. 폴더안으로 다시 들어가 폴더와 파일을 생성/삭제 등 여러 파일 시스템 기능을 수행할 수 있다. 자료구조 시간에 배우는 다양한 클래스와 기능을 사용하여 프로그램을 완성시킨다.

자료구조 시간에 배운 자료 관리 방법들의 장점과 단점을 명확히 알고 단점을 극복할 수 있는 자료 관리 방법을 적용한다. 폴더파일 관리 프로그램을 설계하고 구현한다.

# 프로젝트 목표

* 1. 사진, 음원, 동영상을 관리하기 위한 ‘폴더 시스템’을 구현하는 것이다.
  2. 폴더에 기본적인 기능들을 추가한다.
     1. 최상위 폴더 /root 고정
     2. 하위 폴더 생성가능
     3. 실행 파일 생성 가능
  3. 검색기능은 폴더 안에서 이름으로 찾아 내용이 포함된다면 출력해 준다.
  4. 폴더 목록은 List 자료구조로 설계하며, 각 리스트는 특정한 Primary Key[이름]를 기준으로 하여 정렬되는 Sorted List이다. 각 List의 크기는 5이며, 초과될 시, 5씩 다시 증가해 동적으로 크기를 늘려준다.
  5. 폴더가 추가/삭제될 시에도 Primary Key로 다시 정렬하여 출력해준다.
  6. 하위 폴더로 이동할 시 현재 폴더 중심은 이동한 하위 폴더가 된다. 중심이 된 하위 폴더에서 다시 상위 폴더로 이동할 수 있다. 또한 다시 하위 폴더로 이동할 수 있다(m\_CurPoint(중심)가 바뀌게 된다).
  7. 최근 열어본 폴더를 Queue로 구현한다. 크기는 10이다. 10을 초과할 시 가장 오래전에 열어본 폴더기록을 삭제하고 최근 열어본 정보를 삽입한다.
  8. 프로그램 수행 간 변경되는 사항을 파일 입출력으로 관리할 수 있다.
  9. 현재 폴더 속성을 출력하여 정보를 얻을 수 있다.
  10. 파일을 실행하면 사진(jpg), 음원(mp4), 동영상(avi)에 따라 확장자명이 다르게 적용한다. 최근 접근 시간도 저장되어 출력된다.

# 기능

1. 폴더 관리 프로그램에 요구될 수 있는 기본적인 기능을 포함한다.

* + - 폴더 생성 / 열기 / 삭제 / 검색 / 정렬 / 이름 수정 / 복사
    - 최근 열어본 폴더
    - 현재 폴더 속성
    - 상위폴더로 이동 / 뒤로 가기 / 앞으로 가기

2. 폴더와 파일 관리가 주 목적이다.

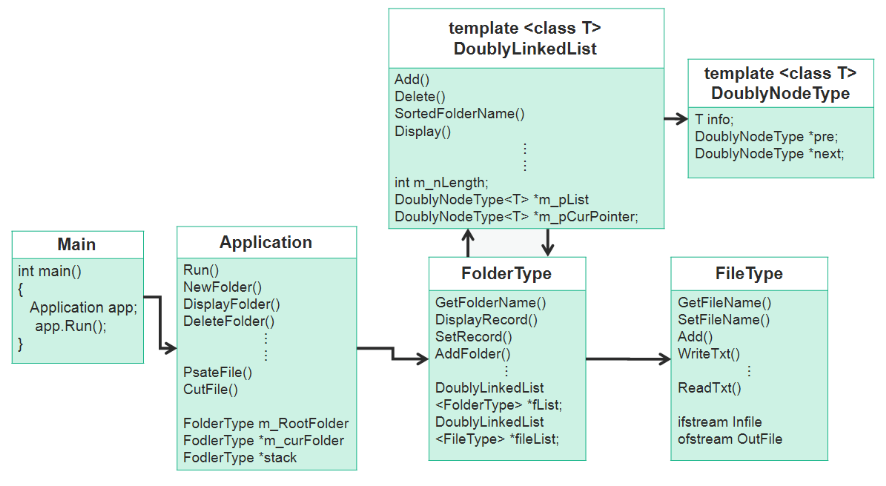
3. 파일 타입에 맞게 파일 제어가 가능하다

- TXT파일이라면 읽기, 쓰기 가능

- 이름 수정 가능

4. GUI를 위한 UI를 개선

# 클래스 다이어그램

****

# ADT

**Class Application**

|  |
| --- |
| <Member functions>  viod Run(); //프로그램 실행 함수  int GetCommand(); //폴더기능 실행  int GetCommand2(); //파일기능 실행  void NewFolder(); //폴더 생성  void DisplayFolder(); //폴더 속 정보 출력  void DeleteFolder(); //폴더 삭제  void MoveSubFolder(); //하위 폴더로 이동  void MoveBack(); //상위 폴더로 이동  void RetrieveFolderByName(); //폴더명으로 검색  void OpenFolderList(); //최근 열어본 폴더정보  void CurrentFolderProperty(); //현재 폴더 속성  void NewFile(); //파일 생성  void RetrieveFileByName(); //파일명으로 검색  void DeleteFile(); //파일 삭제  void FileExecution(); //파일 실행  <Member variables>  FolderType m\_RootFolder; //root폴더  FolderType \*m\_curFolder; //현재 중심 폴더  FolderType \*QueueStack; //상위 폴더로 가기 위한 폴더  int m\_Command; //이 값으로 무엇이 실행할지 결정  int cur = 0; //\*QueueStack의 길이  string s\_Array[10]; //현재 열어본 폴더 값 저장  int s\_cur; //s\_Array의 위치 |

**Class SortedList** template<class T>

|  |
| --- |
| <Member functions>  bool IsFull(); //꽉 차면 동적으로 크기 늘려준다  int Add(T data); //폴더 추가  int Delete(T data); //폴더 삭제  void SortedFolderName(); //폴더명으로 정렬  void Display(T data); //출력  void RetrieveFolder(string temp, T &Ft);//폴더명으로 검색  void RetrieveFile(string temp); //파일명으로 검색  int Overlap(T data); //겹치는 것 있다면 추가 불가능  T& PathExist(T &data); //하위 폴더로 접근하기 위한 함수1  int IntPathExist(T &data); //하위 폴더로 접근하기 위한 함수2  void setPosi(string temp); //위치 저장  void subB(int cur, T &m\_Sub); //QueueStack 쌓는 함수  T& GoB(int cur);  //MoveBack함수에서 상위폴더 주소값 받기  <Member variables>  T \*m\_Array; //하위 폴더 리스트  T \*fd; //QueueStack 쌓는 리스트  int m\_Length; //m\_Array 현재위치  string Path; //위치 저장  int MAXSIZE = 5; //m\_Array 리스트 동적할당 변수 |

**Class FolderType**

|  |
| --- |
| <Member functions>  string GetFolderName(); //Get함수  string GetFolderPosition(); //Get함수  string GetCreatTime() //Get함수  int GetSubFolderNum() //Get함수  bool GetIden() //Get함수  void SetFolderName(string inFolderName) //Set함수  void SetFolderPosition(string inFolderPosition)//Set함수  void SetCreatTime() //Set함수  void SetIden(bool num) //Set함수  void SetSubFolderNum(int inFolderNum) //Set함수  void SetRecord(string inFolderName, string inFolderPosition, string inFolderSize, int inFolderNum)  void DisplayFolderNameOnScreen() //display함수  void DisplayFolderPositionOnScreen() //display함수  void DisplayCreatTimeOnScreen() //display함수  void DisplayFolderNumOnScreen() //display함수  void DisplayRecordOnScreen(FolderType data) //display함수  void SetFolderNameFromKB(); //set 키보드 함수  void SetFolderPositionFromKB(); //set 키보드 함수  void SetRecordFromKB(); //set 키보드 함수  void nullSet() //기본 null로 셋팅  RelationType CompareByID(const FolderType &data);  FolderType& operator= (const FolderType& data);  int AddFolder(); //폴더 추가  int DeleteFolder(); //폴더 삭제  int RetrieveFolderByName(string temp, int num, FolderType &Ft);  //폴더명으로 검색  int RetrieveFileByName(string temp);  //파일명으로 검색  FolderType& Exist(FolderType &data);  //하위 폴더로 접근하기 위한 함수1  int iExist(FolderType data);  //하위 폴더로 접근하기 위한 함수2  void NewfList(string temp) //fList 동적 할당  void subA(int cur, FolderType &m\_Cur) //QueueStack 쌓는 함수  FolderType& GoA(int cur)  //MoveBack함수에서 상위폴더 주소값 받기  void AddFile() //파일 추가  void SortedFileName() //파일명으로 정렬  void DeleteFile() //파일 삭제  void Execution() //파일 실행  <Member variables>  string fName; //폴더이름  string fPath; //폴더위치  string fCreatTime; //생성시간  int fSubFolderNum; //서브폴더 수  SortedList<FolderType> \*fList; //SortedList 리스트  bool iden; //하위폴더 한번 갔었는지 확인하는 변수  FileType \*file; //파일 리스트  int fileNum; //파일 개수  int fileMax; //파일 최대 개수 |

**Class FileType**

|  |
| --- |
| <Member functions>  string GetFileName() //Get 파일명  void SetFileName(string temp) //Set 파일명  string GetFileRP() //Get 파일 연결 프로그램  void SetFileRP(string temp) //Set 파일 연결 프로그램  void SetCreatTime() //파일 생성 시간  void SetAccessTime() //파일 접근 시간  void Add() //파일 정보 추가  void fileShow() //파일 정보 출력  <Member variables>  string fileName; //파일명  string fileRelatedProgram;  //파일 연결프로그램 사진(jpg), 음원(mp4), 동영상(avi)  int fileSize; //파일 크기  string fileCreatTime; //파일 생성시간  string fileAccessTime; //엑세스 시간 |

# 설계 기반 프로그램 및 언어

Visual studio 2017 C++