

File processing prj#1

컴퓨터공학과

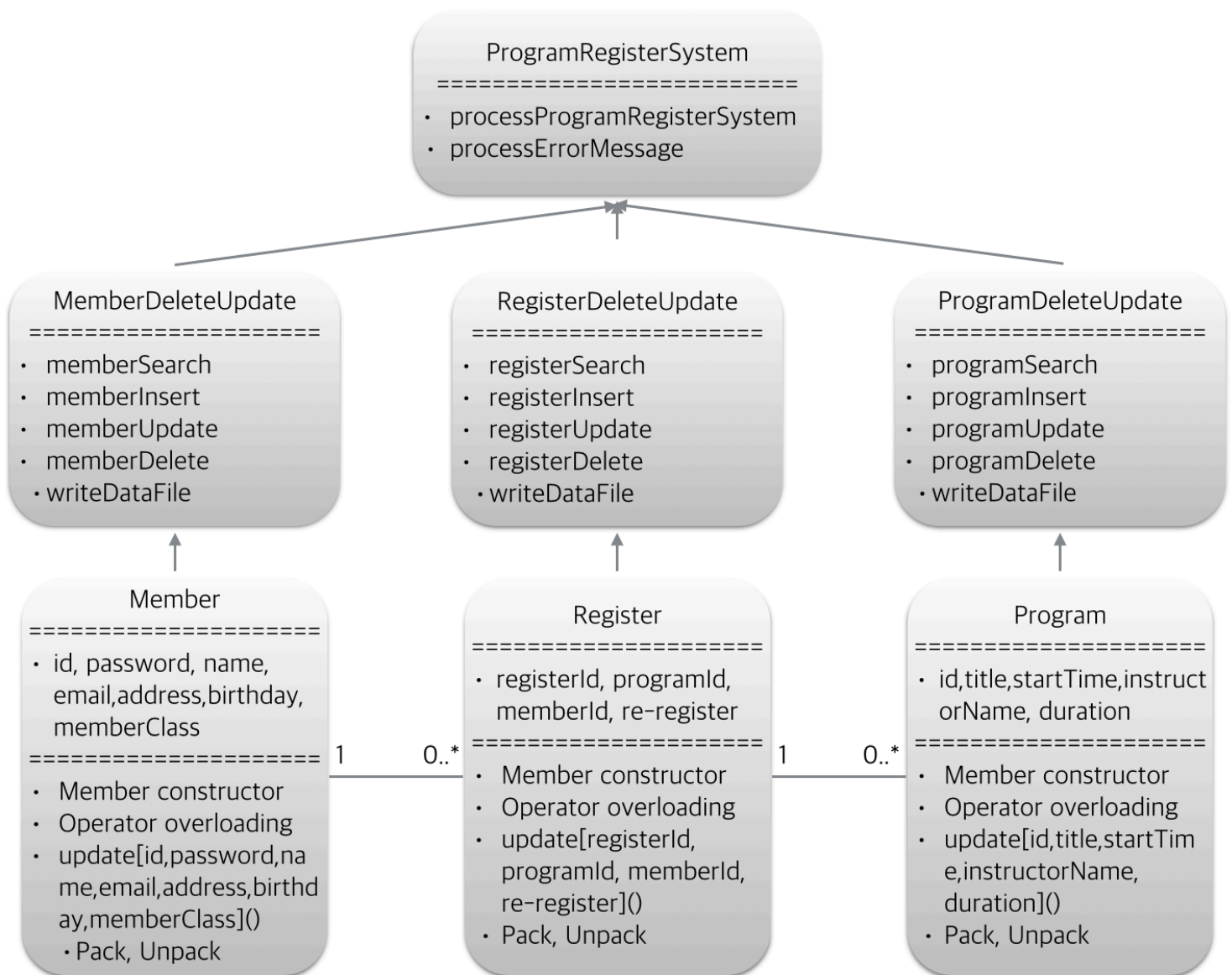
20111596

김상근

1. 프로젝트 진행 중 필요한 가정

- 레코드들은 모두 가변길이 레코드이다.
- 레코드의 필드들은 가변길이 필드로 한다.
- 레코드의 필드들의 타입에 따라 길이를 제한할 수 있다.
- 레코드의 필드들은 ' ' 로 구별한다.
- 모든 경우에서 참조 무결성을 보장하여야 한다.

2. 구현한 Class 의 diagram



3. 각 프로그래밍 문제에서 요구한 사항

- a. 아래와 같은 정보를 처리할 수 있는 Member class 를 작성한다.

Data	Type	Example
ID	Variable string	Batman
Password	Variable string	Bat1234
Name	Variable string	Bruce Wayne
E-mail	Variable string	Batman@wayne.com
Address	Variable string	Gotham City
Birthday	10 chars	1915/02/19
Class	1char	S/A/B/C

- b. 아래와 같은 정보를 처리할 수 있는 Program class 를 작성한다.

Data	Type	Example
ID	8 chars	12345678
Title	Variable string	Swimming
Start Time	10 chars	13:00
Instructor Name	Variable string	Taewhan
Duration	1char	3

- c. 아래와 같은 정보를 처리할 수 있는 Register class 를 작성한다.

Data	Type	Example
Register ID	12 chars	987654321012
Program ID	8 chars	12345678
Member ID	Variable string	Batman
Number of Re-Registered	2 chars	05

- d. 임의의 데이터를 생성한다. (gen[Member,Program,Register].py)
e. IOBuffer 를 이용하여 파일로부터 객체를 메모리에 저장하고 파일에 레코드로서 객체를 저장하는 함수를 정의하고, 각 Class 별로 Test 프로그램을 작성한다.
f. Operator overloading 을 통하여 <<와 >> 를 재정의하고 각 Class 별로 listOfClass.txt 파일의 내용을 출력하는 프로그램을 작성한다.
g. Data 를 Pack 하는 함수를 작성하고, 각각의 정보를 .dat 파일에 저장한다. 이를 Unpack 할 수 있는 함수도 함께 작성한다.

- h. 각각의 레코드를 검색, 삽입, 수정, 삭제하는 기능을 넣어야 한다.
- i. Member 를 삭제했을 때, Register 에 삭제한 member 의 정보도 함께 삭제하는 등의 참조 무결성을 보장하여야 한다.

4. 사용한 파일과 자료구조

- a. member.h : Member class 를 선언하는 부분이다. 생성자와 변수, 메소드 등이 정의되어 있다.
- b. member.cpp : Member class 를 정의하는 부분이다.
- c. program.h : Program class 를 선언하는 부분이다. Member.h 와 유사하다.
- d. Program.cpp : Program class 를 정의하는 부분이다. Member.cpp 와 유사하다.
- e. Register.h : member.h 와 유사하다.
- f. Register.cpp : member.cpp 와 유사하다.
- g. showMember.cpp : listOfMember.txt 에 존재하는 member 를 최대 10 줄의 정보까지 출력해준다.
- h. showProgram.cpp : listOfProgram.txt 에 존재하는 program 을 최대 10 줄의 정보까지 출력해준다.
- i. showRegister.cpp : listOfRegister.txt 에 존재하는 register 를 최대 10 줄의 정보까지 출력해준다.
- j. MemberTest.cpp : listOfMember.txt 의 정보를 읽어온 후, Member class 의 Pack 함수를 이용하여 .dat 파일을 생성한다.
- k. ProgramTest.cpp : listOfProgram.txt 의 정보를 읽어온 후, Program class 의 Pack 함수를 이용하여 .dat 파일을 생성한다.
- l. RegisterTest.cpp : listOfRegister.txt 의 정보를 읽어온 후, Register class 의 Pack 함수를 이용하여 .dat 파일을 생성한다.
- m. Member_manager.h : member class 의 정보를 검색, 삽입, 수정, 삭제할 수 있는 기능에 필요한 메소드들을 선언한 부분이다.
- n. Member_manager.cpp : 위에서 선언한 기능들을 구현하는 부분이고, writeDataFile() 메소드를 구현하여 .dat 파일을 갱신해준다.
- o. Program_manager.h , .cpp: 위 manager 와 유사한 부분이다.
- p. Register_manager.h, .cpp : 위 manager 와 유사한 부분이다.
- q. ProgramRegisterSystem.h : show 함수와 test 함수, 대화형 프로그램에 필요한 메소드를 선언한 부분이다.
- r. ProgramRegisterSystem.cpp : main 함수와 이 프로그램의 Loop 를 정의한 부분이다.

- s. All.h : 본 프로젝트에서 필요한 자료구조들(vector, list 등의 헤더파일들을 전부 포함해 놓은 헤더파일이다.
- t. std::vector<template> 와 같은 동적 배열 자료구조를 이용하여 메모리의 누수없이 효율적으로 코딩하였다.

5. Text book problems

- a. (21) 고정길이 레코드 파일을 사용하는 방식에서 레코드를 삭제할 수 있는 방법은 삭제되었다는 표시로 식별자 '*'를 기록해둔다. 그리고 read 하는 과정에서 '*'가 앞에 있는지 확인하여 삭제되었는지 삭제 되지 않았는지 파악할 수 있다.
- b. (22) 레코드를 추가할 때 삭제된 공간을 활용하기 위하여 가용 리스트를 사용한다. 본 프로젝트는 고정길이 레코드이기 때문에 가용 리스트의 맨 앞에 있는 가용 레코드를 이용하면 된다.
- c. (23) 가변길이 레코드로 변경한다면 우선 모든 삭제된 레코드 위치에 새로운 레코드를 덮어 씌울수가 없어진다. 그리고 해당하는 레코드를 기록해도 새로운 레코드의 길이가 예전 레코드의 길이보다 작을 경우 내용이 겹칠 수도 있다. 따라서 삭제되는 레코드의 크기를 가져와 첫 번째 가용레코드부터 시작하여 크기를 비교해서 현재 레코드의 크기보다 작을 경우 다음 가용 레코드를 불러와서 어떤 것이 더 큰지 비교한다. 이런 방식으로 가용 리스트에 가용 레코드들이 크기 순으로 정렬될 수 있도록 파일을 변경한다.
- d. (24) 새로운 레코드가 입력이 되면 가용 리스트를 탐색하여 자신보다 크기가 큰 레코드 위치가 나오면 해당하는 위치에 삽입한다.
- e. (25) 새 레코드를 삽입하여 update 를 할 때, 가변길이 레코드의 경우 항상 새 레코드와 기존의 레코드의 크기가 같다고 할 수 없다. 새 레코드의 길이보다 기존의 레코드의 길이가 더 긴 경우, 새로운 레코드를 쓰고 남은 부분에 삭제 식별자를 표시한 뒤, 이를 삭제 레코드를 관리하는 리스트에 넣어주어 (24)번의 방식을 이용한다.