File Processing

Programming Project #1

1. Definition and requirement for implementation (80 Point)

본 프로젝트에서는 구립 체육센터의 회원정보와 체육프로그램에 관한 정보를 처리하고 유지하는 정보 시스템을 구축한다.

1.1 Basic class

프로젝트의 기본적인 클래스를 작성한다. 개별 회원에 대한 정보와 개별 체육 프로그램에 대한 정보를 처리하고 유지할 수 있어야 하며, 또한 추후 성능 평가 등의 테스트를 위한 정보 목록을 작성할 수 있어야 한다.

1) 클래스 Member를 작성하라. 각 객체는 한 명의 회원에 대한 정보를 나타낸다. Member 클래스는 **회원의 ID와 비밀번호, 이름, E-mail, 주소, 생년월일, 회원 등급**을 나타내는 데이터를 포함하고, **생성자, 오버로드 된 "="(복사 생성자 및 객체 복사 연산자), "=="(같음 비교 연산자, ID 기준), "!="(다름 비교 연산자, ID 기준), 각 필드 값을 갱신할 수 있는 함수**를 포함한다.

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 characters (Date)	1915/02/19
Class	1 character	S/A/B/C 중 하나

2) 클래스 Program을 작성하라. 각 객체는 하나의 체육 프로그램에 대한 정보를 나타낸다. Program 클래스는 **프로그램의 고유 ID(사이트 관리용), 프로그램 이름, 프로그램 시작 시간, 담당**

강사, 소요 시간을 나타내는 데이터를 포함하고, **생성자, 오버로드 된 "="(복사 생성자 및 객체 복사 연산자), "=="(같음 비교 연산자, ID 기준), "!="(다름 비교 연산자, ID 기준), 각 필드 값을 갱신할 수 있는 함수**를 포함한다.

Data	Type	Example
ID	8 characters (Numeric)	12345678
Title	variable string	Swimming
Start Time	10 characters (Time)	13:00
Instructor Name	variable string	Taehwan
Duration	1 character (Numeric)	3

3) 클래스 Register를 작성하라. 각 객체는 하나의 프로그램 등록에 대한 정보를 나타낸다. Register 클래스는 **등록 고유 ID(사이트 관리용), 프로그램의 고유 ID(Program 클래스의 ID와 동일), 등록 회원의 ID(Member 클래스의 ID와 동일), 재등록 횟수**를 나타내는 데이터를 포함하고, **생성자, 오버로드 된 "="(복사 생성자 및 객체 복사 연산자), "=="(같음 비교 연산자, ID 기준), "!="(다름 비교 연산자, ID 기준), 각 필드 값을 갱신할 수 있는 함수**를 포함한다.

Data	Type	Example
Register ID	12 characters (Numeric)	987654321012
Program ID	8 characters (Numeric)	12345678
Member ID	variable string	batman
Number of Re-registered	2 characters (Numeric)	05

4) 실제 회원 목록과 프로그램 목록, 등록 목록을 작성하라. 특히, **등록 목록의 데이터는 작성한 회원 목록과 프로그램 목록에 존재하는 데이터로 구성**되어야 한다. 회원 목록과 프로그램 목록은 최소 각각 1,000개의 레코드를, 등록 목록은 최소 10,000개의 레코드를 작성 해야 한다. 데이터는 실제 존재하는 데이터를 사용할 필요는 없으며, **임의로 생성해도 상관 없지만 앞에서 보인 예제 형식을 따라야** 한다. 데이터 작성의 편의상 본 프로젝트에서는 **Program과 Member는 다 대 다의 관계**를 가지고, **Member와 Register, Program과 Register는 1 대 다**의 관계를 가진다. 각자 작성한 데이터들을 종합하여 프로그램의 성능을 평가하는데 사용하기 때문에 **데이터가 입력되는 과정에서 중복 되는 경우가 발생할 수 있으므로** 중복되는 레코드가 발생하지 않도록 모든 경우에 이를 염두에 두고 프로그램을 작성한다. 각각의 파일 이름은 **listOfMember.txt**, **listOfProgram.txt**, **listOfRegister.txt**로 한다 각각의 필드를 구분자 **'|'**(vertical bar, a.k.a. pipe)를 사용하여 구분하며 첫 줄에는 레코드의 총 개수를 그 다음 줄부터는 각 줄에 하나의 레코드를 나타낸다. 다음 예시를 참고하여 데이터를 작성 하도록 한다.

```
listOfMember.txt
1000
batman|bat1234|Bruce Wayne|batman@wayne.com|Gotham City|1915/02/19|S
...

listOfProgram.txt
1000
12345678|Swimming|13:00|Taehwan|3
...

listOfRegister.txt
10000
987654321012|12345678|batman|05
...
```

1.2 Adding methods to basic classes

입력 스트림에서 객체를 읽고 출력 스트림으로 형식화 된 객체를 내보내기 위해 클래스에 함수를 추가한다.

1) 입력 스트림으로부터 회원 필드 값을 읽고 출력 스트림에 필드 값을 출력하기 위해 연산자 오버로딩 함수를 추가하라. 또한 클래스가 정확히 구현 된다는 것을 확인할 수 있는 테스트 프로그램을 작성하라. 테스트 프로그램은 1.1.4에서 작성한 레코드들을 읽은 뒤 표준 출력으로 앞의 10개의 레코드만 출력 할 수 있도록 한다. 다음을 참고하여 구현할 것.

- `istream & operator>> (istream & is, Member & m);`

  1.1.4의 예시와 같은 입력 스트림으로 부터 데이터를 입력 받아 객체에 저장

- `ostream & operator<< (ostream & os, Member & m);`

  객체 데이터를 적절한 형태로 출력 스트림으로 출력

- 테스트 프로그램의 이름은 **showMember**로 할 것

2) 입력 스트림으로부터 앨범 필드 값을 읽고 출력 스트림에 필드 값을 출력하기 위해 연산자 오버로딩 함수를 추가하라. 또한 클래스가 정확히 구현 된다는 것을 확인할 수 있는 테스트 프로그램을 작성하라. 테스트 프로그램은 1.1.4에서 작성한 레코드들을 읽은 뒤 표준 출력으로 앞의 10개의 레코드만 출력 할 수 있도록 한다. 다음을 참고하여 구현할 것.

- `istream & operator>> (istream & is, Program & m);`

1.1.4의 예시와 같은 입력 스트림으로 부터 데이터를 입력 받아 객체에 저장

- `ostream & operator<< (ostream & os, Program & m);`

  객체 데이터를 적절한 형태로 출력 스트림으로 출력

- 테스트 프로그램의 이름은 **showProgram**으로 할 것

3) 입력 스트림으로부터 구매 필드 값을 읽고 출력 스트림에 필드 값을 출력하기 위해 연산자 오버로딩 함수를 추가하라. 또한 클래스가 정확히 구현 된다는 것을 확인할 수 있는 테스트 프로그램을 작성하라. 테스트 프로그램은 1.1.4에서 작성한 레코드들을 읽은 뒤 표준 출력으로 앞의 10개의 레코드만 출력 할 수 있도록 한다. 다음을 참고하여 구현할 것.

- `istream & operator>> (istream & is, Register & m);`

  1.1.4의 예시와 같은 입력 스트림으로 부터 데이터를 입력 받아 객체에 저장

- `ostream & operator<< (ostream & os, Register & m);`

  객체 데이터를 적절한 형태로 출력 스트림으로 출력

- 테스트 프로그램의 이름은 **showRegister**로 할 것

1.3 Using IOBuffer

교재 4장에 수록된 IOBuffer를 사용하여 파일로부터 객체를 메모리에 적재하고 파일에 레코드로써 객체를 저장하는 함수를 추가한다.

1) 클래스 Member에 Pack과 Unpack 함수를 추가하라. 회원 레코드의 파일을 생성하기 위해 교재에 수록된 클래스 BufferFile을 사용한다. 또한 IOBuffer 클래스에 의해 제공되는 버퍼 타입을 사용하여 이 함수를 테스트 하는 프로그램을 구현하라. 다음을 참고하여 구현할 것.

- 1.1.4에서 작성한 데이터를 읽고 Pack 함수를 사용하여 **fileOfMember.dat** 파일을 작성한 후, 작성 된 fileOfMember.dat 파일에서 Unpack 함수를 사용하여 저장된 레코드들을 읽은 뒤 표준 출력으로 출력하도록 한다.

- 테스트 프로그램의 이름은 **MemberTest**로 할 것. 테스트 프로그램의 실행 후 fileOfMember.dat 파일이 생성되어야 하며 입력되는 데이터 파일의 모든 레코드를 저장하여야 한다. 다만 Unpack을 테스트하기 위해 표준 출력으로 출력되는 레코드들은 앞의 10

개만 출력하도록 한다.

2) 클래스 Program에 Pack과 Unpack 함수를 추가하라. 앨범 레코드의 파일을 생성하기 위해 교재에 수록된 클래스 BufferFile을 사용한다. 또한 IOBuffer 클래스에 의해 제공되는 버퍼 타입을 사용하여 이 함수를 테스트 하는 프로그램을 구현하라. 다음을 참고하여 구현할 것.

- 1.1.4에서 작성한 데이터를 읽고 Pack 함수를 사용하여 **fileOfProgram.dat** 파일을 작성한 후, 작성 된 fileOfProgram.dat 파일에서 Unpack 함수를 사용하여 저장된 레코드들을 읽은 뒤 표준 출력으로 출력하도록 한다.

- 테스트 프로그램의 이름은 **ProgramTest**로 할 것. 테스트 프로그램의 실행 후 fileOfProgram.dat 파일이 생성되어야 하며 입력되는 데이터 파일의 모든 레코드를 저장하여야 한다. 다만 Unpack을 테스트하기 위해 표준 출력으로 출력되는 레코드들은 앞의 10개만 출력하도록 한다.

3) 클래스 Register에 Pack과 Unpack 함수를 추가하라. 예매 레코드의 파일을 생성하기 위해 교재에 수록된 클래스 BufferFile을 사용한다. 또한 IOBuffer 클래스에 의해 제공되는 버퍼 타입을 사용하여 이 함수를 테스트 하는 프로그램을 구현하라. 다음을 참고하여 구현할 것.

- 1.1.4에서 작성한 데이터를 읽고 Pack 함수를 사용하여 **fileOfRegister.dat** 파일을 작성한 후, 작성 된 fileOfRegister.dat 파일에서 Unpack 함수를 사용하여 저장된 레코드들을 읽은 뒤 표준 출력으로 출력하도록 한다.

- 테스트 프로그램의 이름은 **RegisterTest**로 할 것. 테스트 프로그램의 실행 후 fileOfRegister.dat 파일이 생성되어야 하며 입력되는 데이터 파일의 모든 레코드를 저장하여야 한다. 다만 Unpack을 테스트하기 위해 표준 출력으로 출력되는 레코드들은 앞의 10개만 출력하도록 한다.

1.4 Deleting and Updating Records

파일로부터 레코드를 삭제하고 파일에 있는 레코드를 수정할 수 있는 함수를 추가한다. 이 문제는 교재에 수록된 연습문제 6장의 21 ~ 25번을 참조할 것.

1) 교재 **6장의 연습문제 21 ~ 25**에서 기술된 Delete와 Update 연산을 참조 및 이용하여 삭제와 수정을 지원하는 회원, 앨범, 구매 레코드 파일을 생성하라. 또한 각각의 파일에서 회원, 앨범, 구매 레코드의 **검색, 삽입, 삭제, 수정을 지원할 수 있는 대화식 프로그램**을 작성하라. 수정은 키(ID)를 제외한 모든 필드가 가능하며, 모든 경우에서 **참조 무결성을 유지**해야 한다.

- 참조 무결성은, 예를 들어, 회원 레코드가 삭제되면 그 회원이 구매한 기록도 모두 같이 삭제 될 수 있도록 하는 것 등을 말한다.

- 각각의 레코드 파일의 이름은 1.3에서와 같이 **fileOfMember.dat, fileOfProgram.dat, fileOfRegister.dat**으로 한다. 즉, **최종 제출 버전의 프로그램에서는 삭제와 수정 연산을 지원하는 파일을 작성**해야 한다.

- 대화식 프로그램의 이름은 **ProgramRegisterSystem**으로 할 것.

- **6장 연습문제 21 ~ 25번에 대한 답을 문서에 반드시 정리할 것.**

- 자신이 채택한 검색, 삽입, 삭제, 수정 방법과 그 방법이 효율적인 이유를 문서에 정리할 것.

- 삭제에 따른 파일 compaction 전략을 문서에 기술하고 직접 구현할 것.

- 레코드 검색 : record의 ID로 검색. Register의 경우 Member ID나 Program ID를 사용한 검색도 지원해야 한다.

- 레코드 삽입 : record의 ID를 사용하여 검색 후 삽입

- 레코드 삭제 : record의 ID를 사용하여 검색 후 삭제

- 레코드 수정 : record의 ID를 사용하여 검색 후 수정

- **반드시 C++ Object-Oriented Programming을 할 것**

2. Requirement for Document (20 Point)

1) 프로젝트 진행 중 자신이 필요하다고 생각되는 가정은 반드시 문서에 기술할 것

2) 구현한 클래스의 다이어그램(standard UML specification)을 그릴 것.

3) 각 프로그래밍 문제에서 요구한 사항을 빠짐없이 기술할 것.

4) 최대한 자세히 기술하되, 소스 코드는 포함하지 말 것.

5) 각 파일과 자료구조 등에 대한 설명을 포함할 것.

3. Due Date and Submission

1) 기한 : **2014.10.27(월) 17:00까지**

2) 제출 방식

- E-mail : **filepro2014@gmail.com**

  제목과 파일 이름을 다음과 같은 형식에 맞추어서 제출

  FP_PP_1_학번_이름 (예 : FP_PP_1_20141234_홍길동)

- hard copy : document를 출력하여 AS915 앞 상자에 제출. 반드시 학번과 이름을 명시할 것.

3) 제출 양식

- 첨부파일은 document와 프로젝트 파일을 같이 압축하여 제출한다. document는 압축 파일의 root에 있어야 한다.

- 제출파일은 반드시 zip 형식으로 압축한다. (예 : FP_PP_1_20141234_홍길동.zip)

- document는 한컴 한글이나 MS Word로 작성하며 **파일 제목은 메일의 제목과 같은 양식**을 지킨다. (예 : FP_PP_1_20141234_홍길동.doc)

- 프로젝트는 반드시 **MS Visual Studio 2010**에서 **Win32 콘솔 응용 프로그램**으로 진행하며, 제출할 때 **build clean**을 꼭 수행하고 **debug, ipch 폴더와 sql 파일을 삭제**한 후 압축하여 제출한다. **프로젝트의 이름도 메일 제목과 같은 양식**을 따른다. (예 : FP_PP_1_20141234_홍길동)

- 메일을 다시 제출하는 경우 메일 제목의 끝에 반드시 (재 제출) 이라고 명시한다.

- 제출 형식을 지키지 않는 경우 감점의 사유가 될 수 있다.

4. Assessment

1) 배점에 따라 기능 구현 여부, error handling, 대화형 방식의 적절성을 평가

2) 문서는 각 요구사항을 상, 중, 하로 구분하여 평가

3) 다음과 같은 경우 감점

- 기한을 지키지 않은 경우. 당일 24시까지 제출하는 경우 20% 감점, 하루 late는 40% 감점, 이틀 late는 70%감점하고 그 이후는 미 제출 처리

- E-mail이나 hard copy 중 한 방식으로만 제출하면 50% 감점

- 테스트 도중 segmentation fault 등의 비 정상적인 종료가 발생하는 경우 50% 감점

4) 다음과 같은 경우 0점

- C++이 아닌 언어를 사용하거나, 객체 지향 프로그래밍을 사용하지 않은 경우

- 첨부 파일에 바이러스가 있는 경우

- 첨부 파일의 압축을 풀 수 없는 경우

5) 소스 코드 혹은 문서의 내용을 다른 사람과 Copy하는 경우 무조건 0점 처리하며, 이 후 두 번째 Copy가 적발되면 성적에 관계없이 F학점

5. Announcement

1) 각 클래스는 앞으로 진행될 프로젝트에서 계속 사용되므로 반드시 재사용성을 고려할 것

2) 질문 사항은 과목 게시판 (데이터베이스 연구실 홈페이지 게시판)을 이용할 것

3) 프로젝트 진행 중 발생한 모든 문제점에 대해 최대한 노력을 기울여 해결하고, 만약 해결할 수 없다면 문제점에 대해 논리적으로 문서에 상세히 기술할 것.

4) 제출하는 프로그램은 하나의 프로그램으로 제작 되어야 하며, 앞에서 요구한 각각의 프로그램들은 하나의 프로그램 안의 부 프로그램으로 구성되어서 메뉴 선택을 통해 원하는 프로그램을 실행할 수 있어야 한다. 프로그램의 구성은 각자의 자유로 맡기지만 최대한 사용자의 편의성을 고려할 것.