**파일처리 프로젝트#1**

**[인터넷 강의 회원관리 시스템]**

**20131587**

**이수영**

1. **프로젝트 목적**

본 프로젝트에서는 인터넷 강의의 회원 관리를 위해 회원 정보와 강의 수강 내역에 관한 정보를 처리하고 유지하는 정보 시스템을 구축한다.

1. **프로젝트 기본 요건**

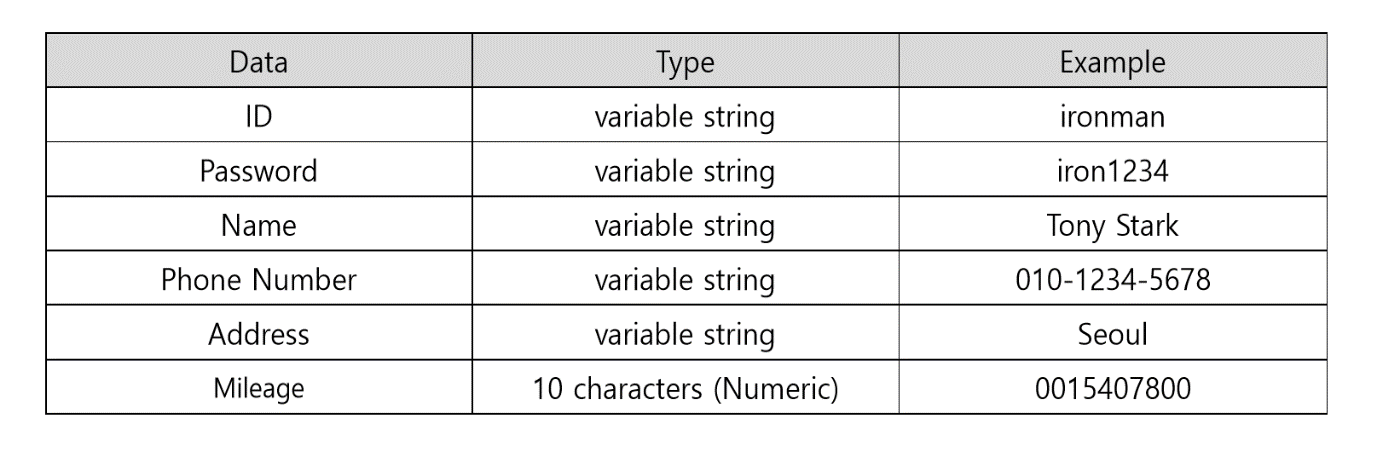
2-1. listOfMember, listOfLecture, listOfPurchase (.txt) 파일들은 무결성이 보장되어 있어야 한다.

2-2. 프로그램 실행 후 LecturePurchaseSystem 을 처음 작동 시킬 때에 data 파일이 생성된다.

1. **각 클래스에 관한 설명(자료구조 (Field), Attribute, Method 등)**

**<Member>**

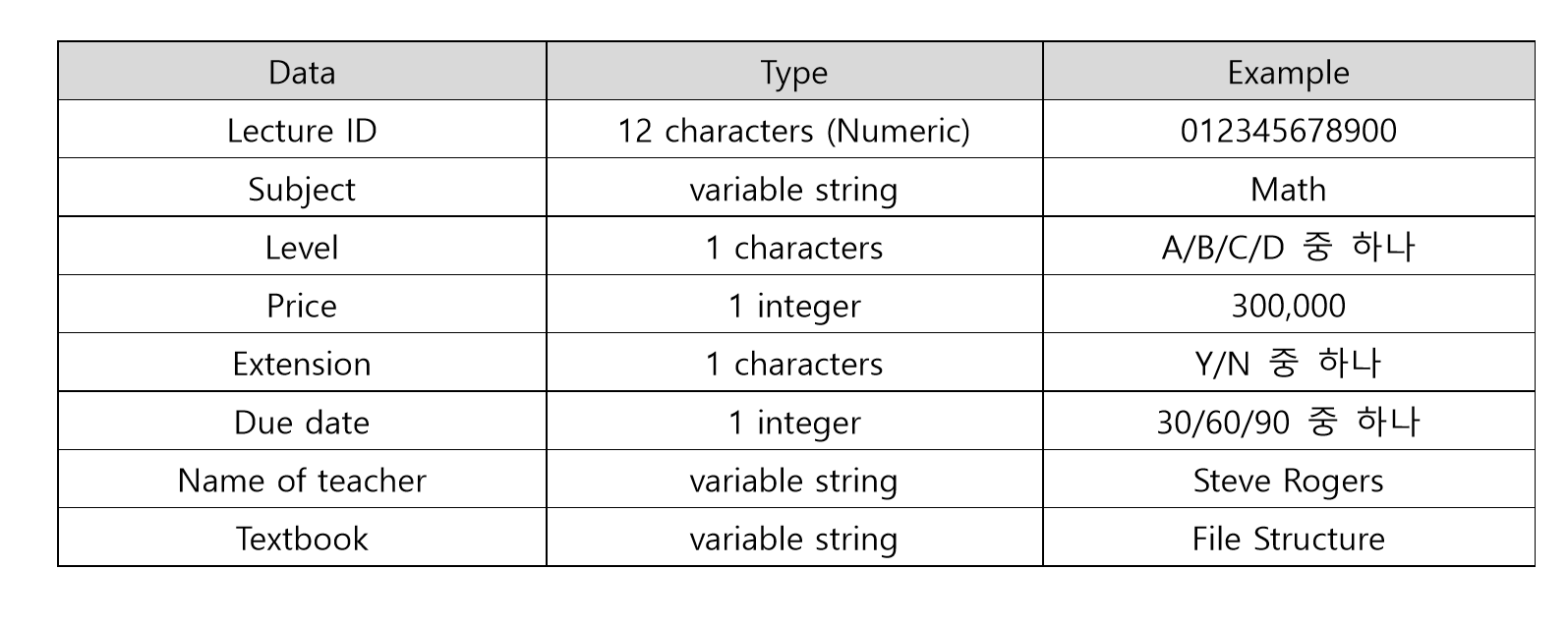
이 객체는 한 명의 회원에 대한 정보를 나타낸다. ID, Password, Name, Phone Number, Address, Mileage등의 private field와 이 객체를 생성할 생성자, field들을 수정할 method, object 비교/대입 연산에 사용할 Overloading된 연산자로 구성되어 있다. 각 field들에 대한 정보는 다음과 같다.



앞서 언급한 method 이외에도, Pack, Unpack method와 object 전체의 길이를 반환해주는 method, object의 출력과 입력에 사용할 method 또한 구현되어 있다.

**<Lecture>**

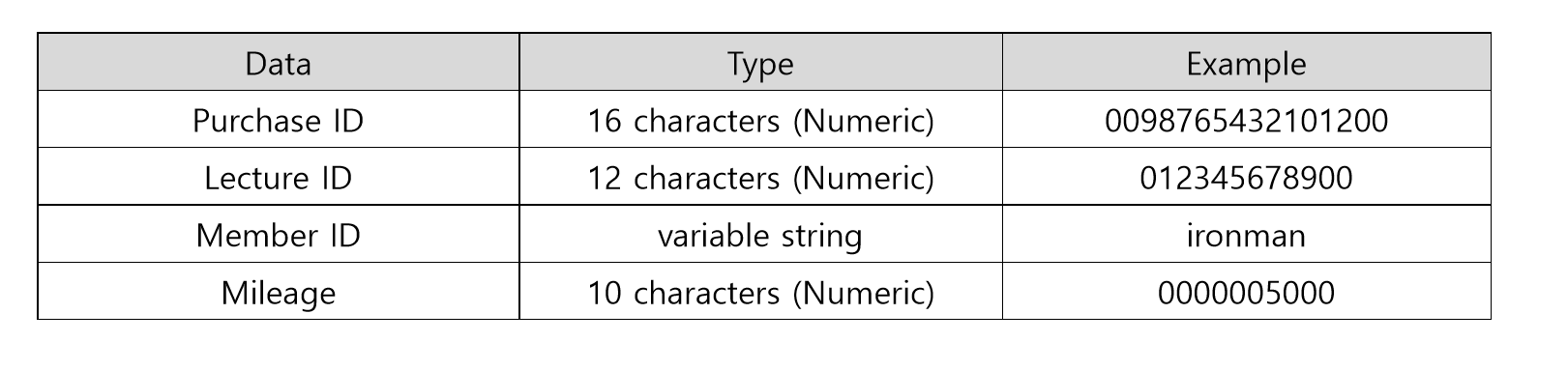
이 객체는 하나의 Lecture에 대한 정보를 나타낸다. Lecture ID, Subject, Level Price, Extension, Due date, Name of teacher, Textbook 등의 private field와 이 객체를 생성할 생성자, field 들을 수정할 method, object 비교/대입 연산에 사용할 오버로드 된 연산자로 구성되어 있다. 각 field 들에 대한 정보는 다음과 같다.



앞서 언급한 method 이외에도, Pack, Unpack method와 object 전체의 길이를 반환해주는 method, object의 출력과 입력에 사용할 method 또한 구현되어 있다.

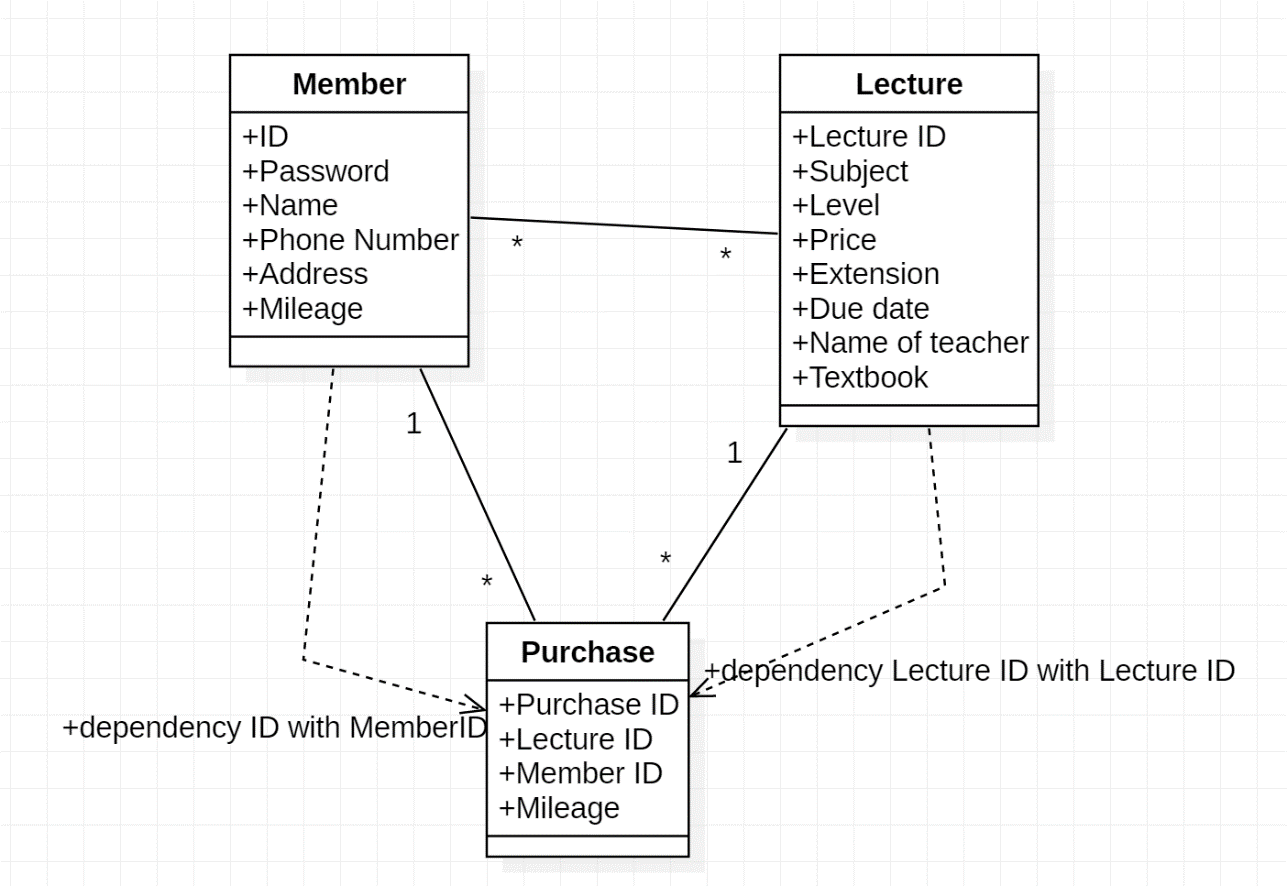
**<Purchase>**

이 객체는 하나의 수강 내역에 대한 정보를 나타낸다. Purchase ID, Lecture ID, Member ID, Mileage 등의 private field와 이 객체를 생성할 생성자, field 들을 수정할 method, object 비교/대입 연산에 사용할 오버로드 된 연산자로 구성되어 있다. 각 field 들에 대한 정보는 다음과 같다.



앞서 언급한 method 이외에도, Pack, Unpack method와 object 전체의 길이를 반환해주는 method, object의 출력과 입력에 사용할 method 또한 구현되어 있다.

1. **구현한 클래스의 다이어그램(standard UML specification)**



Lecture 과 Member는 다 대 다의 관계를 가지고, Member와 Purchase, Lecture 와 Purchse는 1 대 다의 관계를 가진다. 참조 무결성을 유지하기 위해, 회원레코드가 삭제되면 회원 ID를 참조하는 Purchase의 정보도 사라진다. 또한, Purchase의 레코드가 추가 될 때, Member ID나 Lecture ID 가 각각 Member 의 ID 나 Lecture의 ID에 존재 하지 않으면, 이는 추가 될 수 없다.

1. **함수, 메소드 추가구현 사항**

**<showMember>**

: 입력 스트림으로부터(listOfMember.txt) Member의 Field 값을 읽고 출력 스트림에 Field 값을 출력하기 위해 추가한 오버로딩 함수를 추가한 사실을 확인하기 위한 테스트 프로그램이다. 작성된 레코드들을 읽은 뒤 표준 출력으로 앞의 10개의 레코드만 출력한다.

**<showLecture>**

입력 스트림으로부터(listOfLecture.txt) Lecture의 field 값을 읽고 출력 스트림에 field값을 출력하기 위해 추가한 오버로딩 함수를 추가한 사실을 확인하기 위한 테스트 프로그램이다. 작성된 레코드들을 읽은 뒤 표준 출력으로 앞의 10 개의 레코드만 출력한다.

**<showPurchase>**

입력 스트림으로부터(listOfPurchase.txt) Purchase의 field 값을 읽고 출력 스트림에 field 값을 출력하기 위해 추가한 오버로딩 함수를 추가한 사실을 확인하기 위한 테스트 프로그램이다. 작성된 레코드들을 읽은 뒤 표준 출력으로 앞의 10 개의 레코드만 출력한다.

**<MemberTest>**

Member 클래스에 추가한 Pack과 Unpack 함수를 확인하기 위한 Test Program이다. listOfMember.txt로부터 데이터를 읽어 들인 후, Pack(IOBuffer)함수를 사용하여 fileOfMember.dat 파일을 작성한다. 작성된 fileOfMember.dat 파일에서 Unpack 함수를 사용하여 저장된 레코드들을 읽은 뒤 표준출력으로 앞의 10개의 레코드만 출력한다.

**<LectureTest>**

Lecture 클래스에 추가한 Pack과 Unpack 함수를 확인하기 위한 Test Program이다. listOfLecture.txt로부터 데이터를 읽어 들인 후, Pack(IOBuffer)함수를 사용하여 fileOfLecture.dat 파일을 작성한다. 작성된 fileOfLecture.dat 파일에서 Unpack 함수를 사용하여 저장된 레코드들을 읽은 뒤 표준출력으로 앞의 10개의 레코드만 출력한다.

**<PurchaseTest>**

Purchase 클래스에 추가한 Pack과 Unpack 함수를 확인하기 위한 Test Program이다. listOfPurchase.txt로부터 데이터를 읽어 들인 후, Pack(IOBuffer)함수를 사용하여 fileOfPurchase.dat 파일을 작성한다. 작성된 fileOfPurchase.dat 파일에서 Unpack 함수를 사용하여 저장된 레코드들을 읽은 뒤 표준출력으로 앞의 10개의 레코드만 출력한다.

**<LecturePurchaseSystem>**

각 데이터 파일(fileOfMember.dat, fileOfLecture.dat, fileOfPurchase.dat)을 통해서 검색, 삽입, 삭제, 수정을 지원하는 대화식 프로그램. 검색(Search)/삽입(Insert)/삭제(Delete)/수정(Modify)의 과정들은 각각의 ID들을 통해 이루어진다. 모든 경우에서 참조 무결성을 유지해야 하며, 수정에서는 ID를 제외한 모든 필드가 수정 가능하다.

**<Search>**

Record의 ID로 검색 후 검색한 레코드를 출력한다. Purchase의 경우 Purchase ID 뿐만 아니라 Member ID나 Lecture ID를 통한 검색도 지원 한다. 해당 함수를 호출할 때 플래그 값을 두어서 다른 함수에서 특정 ID에 맞는 Record를 검색해야 할 때 사용할 수 있게 하였다.(Reuse)

**Search에 적용한 방법**

: 아직 index나 RRN을 추가하지 않았기 때문에 일단 순차 검색을 통해 해당 ID에 관련된 Record들을 검색한다.

**<Insert>**

Record의 ID를 사용하여 검색 후, 만약 이미 존재하는 ID라면 해당 메시지를 출력 후 LecturePurchaseSystem menu로 돌아가고, 만약 아직 존재하지 않는 ID라면 해당 record에 대한 삽입을 진행할 수 있게 해준다. Insert 과정에서 형식에 맞지 않는 data를 입력하려 하면 바로 출력을 해준다.

**Insert에 적용한 방법**

: 위의 Search 함수를 활용하여 해당 ID를 순차적으로 탐색하고, 이를 찾지 못했을 때, 해당 ID에 맞는 데이터들을 삽입할 수 있게 한다.

**<Delete>**

Record의 ID를 사용하여 검색 후, 참조 무결성을 유지하면서 해당 ID에 대한 record들을 삭제한다.

**Delete에 적용한 방법**

: 위의 Search 함수를 활용하여 해당 ID를 순차적으로 탐색하고, 이를 찾았을 때 해당 ID와 관련된 모든 Record들을 삭제한다.(참조 무결성을 유지하기 위함.)

**삭제에 따른 파일 Compatcion 전략**

: 순차적으로 탐색 후, 해당 ID에 관련된 record를 삭제했을 때는 그 공간 만큼을 Shift 해서 메모리 중간의 빈 공간을 없앤다. 추후 삽입 시에는 record의 맨 뒤로 삽입한다. 현재 수업시간에 배운 내용으로는 레코드를 다룰 때, 길이 지시자를 이용하는 방법과 각 레코드의 끝에 구분자를 놓는 방법, 주소를 기억하는 인덱스를 사용하는 방법 등을 제시했는데, 본 프로젝트에서는 가변 길이 문자열을 사용하는 필드가 많았고, 따라서 길이에 따른 검색이 어려웠다. 또한, 인덱스를 사용하기도 위와 같은 이유로 인해 어려웠다. 결과적으로, 현재는 Shift하는 방식으로 구현한 것이 최선이라 여겨지며, 추후에 프로젝트가 진행되며 이 부분에 대한 성능 향상을 고려해볼 수 있을 것이다.

**<Modify>**

Record의 ID 를 사용하여 검색 후, 아직 존재하지 않는 ID라면 해당 메시지를 출력 후, LecturePurchaseSystem menu로 돌아가고, 이미 존재하는 ID 라면 해당 record에 대한 정보를 수정할 수 있게 해준다. Modify 과정에서 형식에 맞지 않는 data를 입력하면 에러임을 바로 알려준다.

1. **사용한 파일에 대한 설명**

**[main.cpp]**

Document를 통해 제시된 부 프로그램들과 이를 통합하는 Main 프로그램에 구현되어 있다. 이는 다음과 같은 함수들로 이루어져있다.

showMenu() : Main menu를 보여주는 함수.

showMember() : Document를 통해 제시된 showMember를 구현

showLecture() : Document를 통해 제시된 showLecture를 구현

showPurchase() : Document를 통해 제시된 showPurchase를 구현

MemberTest(int print\_flag) : Document를 통해 제시된 MemberTest를 구현

LectureTest(int print\_flag) : Document를 통해 제시된 LectureTest를 구현

PurchaseTest(int print\_flag) : Document를 통해 제시된 PurchaseTest를 구현

LecturePurchaseSystem()

: Document를 통해 제시된 LecturePuchaseSystem의 뼈대를 이루는 함수

showSystemMenu() : LecturePuchaseSystem menu를 보여주는 함수.

SearchingByID(int flag, string filename, string ID)

: Document를 통해 제시된 Search을 구현

InsertingByID() : Document를 통해 제시된 Insert를 구현

DeletingByID() : Document를 통해 제시된 Delete를 구현

ModifyingByID() : Document를 통해 제시된 Modify를 구현

DeleteMember(string ID) : 특정 아이디와 관련된 MemberRecord를 삭제

DeleteLecture(string ID) : 특정 아이디와 관련된 LectureRecord를 삭제

DeletePurchase(string ID) : 특정 아이디와 관련된 PurchaseRecord를 삭제

**[Member.h]**

Member Class의 Field 자료구조 및 각 Field의 setter와 getter가 구현되어 있다. 또한 해당 객체의 전체 길이를 반환해주는 함수도 구현되어 있다.

**[Member.cpp]**

Member Class의 객체간의 비교/복사 연산에 사용될 오버로드된 연산자와 Pack/Unpack method 및 object의 출력과 입력에 사용될 method가 구현되어 있다.

**[Lecture.h]**

Lecture Class의 Field 자료구조 및 각 Field의 setter와 getter가 구현되어 있다. 또한 해당 객체의 전체 길이를 반환해주는 함수도 구현되어 있다.

**[Lecture.cpp]**

Lecture Class의 객체간의 비교/복사 연산에 사용될 오버로드된 연산자와 Pack/Unpack method 및 object의 출력과 입력에 사용될 method가 구현되어 있다.

**[Purchase.h]**

Purchase Class의 Field 자료구조 및 각 Field의 setter와 getter가 구현되어 있다. 또한 해당 객체의 전체 길이를 반환해주는 함수도 구현되어 있다.

**[Purchase.cpp]**

Purchase Class의 객체간의 비교/복사 연산에 사용될 오버로드된 연산자와 Pack/Unpack method 및 object의 출력과 입력에 사용될 method가 구현되어 있다.