**파일처리 프로젝트#2**

**[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. 여기에 더해, 앞의 프로젝트에서 구현한 내용을 TextIndex, TextIndexBuffer, TextIndexedFile 을 이용하여 개선하고, 기존의 시스템에 관리자 모드 및 일반 사용자 모드를 추가하여 access control을 진행하고, 인덱스 기능을 추가한다.

1. **프로젝트 기본 요건**

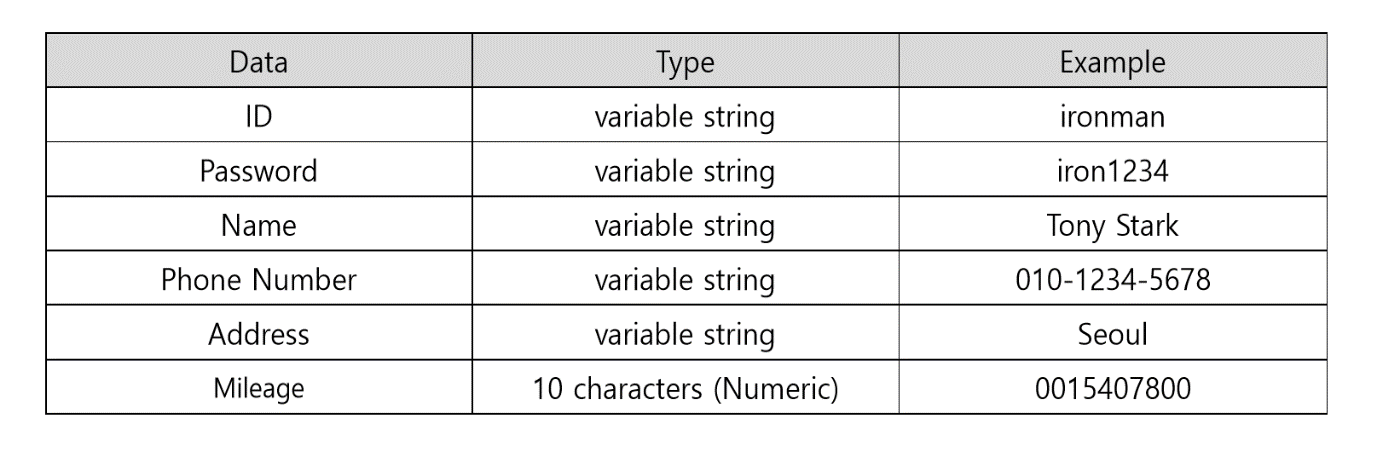
2-1. listOfMember, listOfLecture, listOfPurchase (.txt) 파일들은 무결성이 보장되어 있어야 한다.

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>**

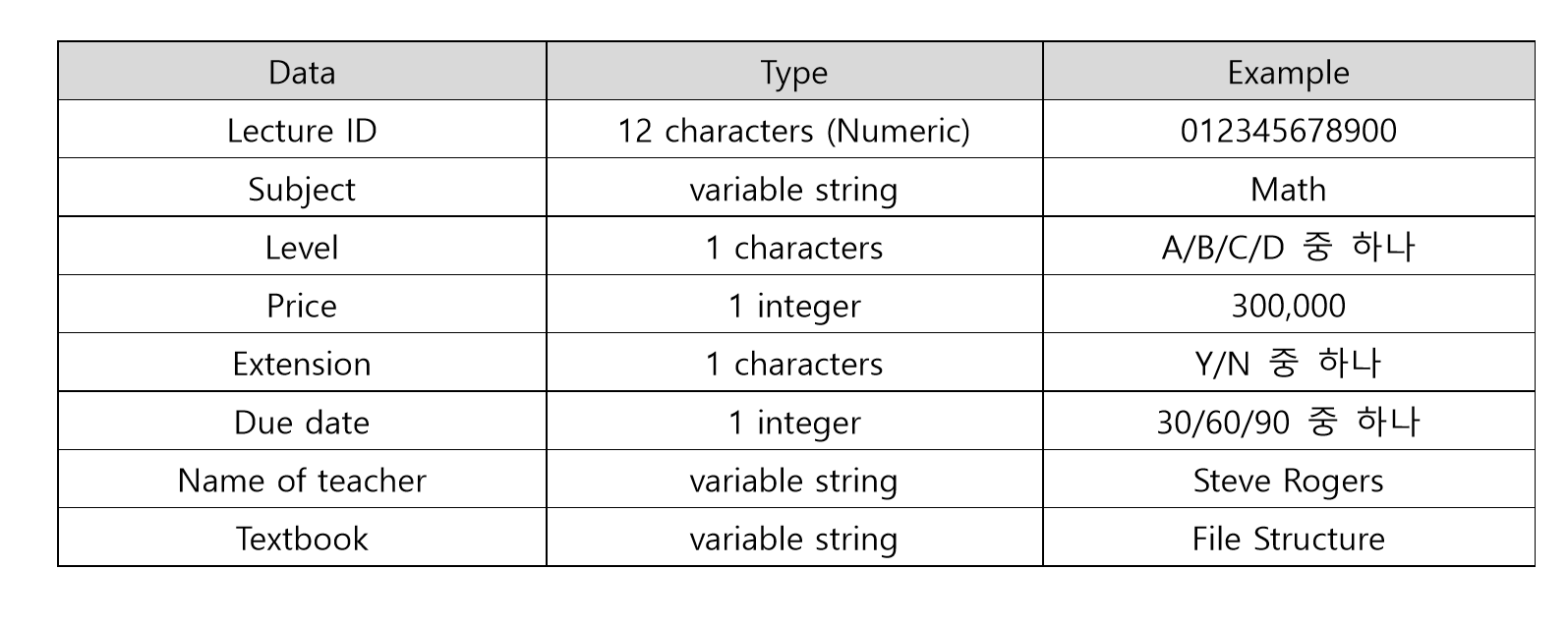
이 객체는 한 명의 회원에 대한 정보를 나타낸다. ID, Password, Name, Phone Number, Address, Mileage등의 private field와 이 객체를 생성할 생성자, field들을 수정할 method, object 비교/대입 연산에 사용할 Overloading된 연산자로 구성되어 있다. 각 field들에 대한 정보는 다음과 같다. 기존의 프로젝트에 더해 level과 key 필드가 추가되었다.



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

**<Lecture>**

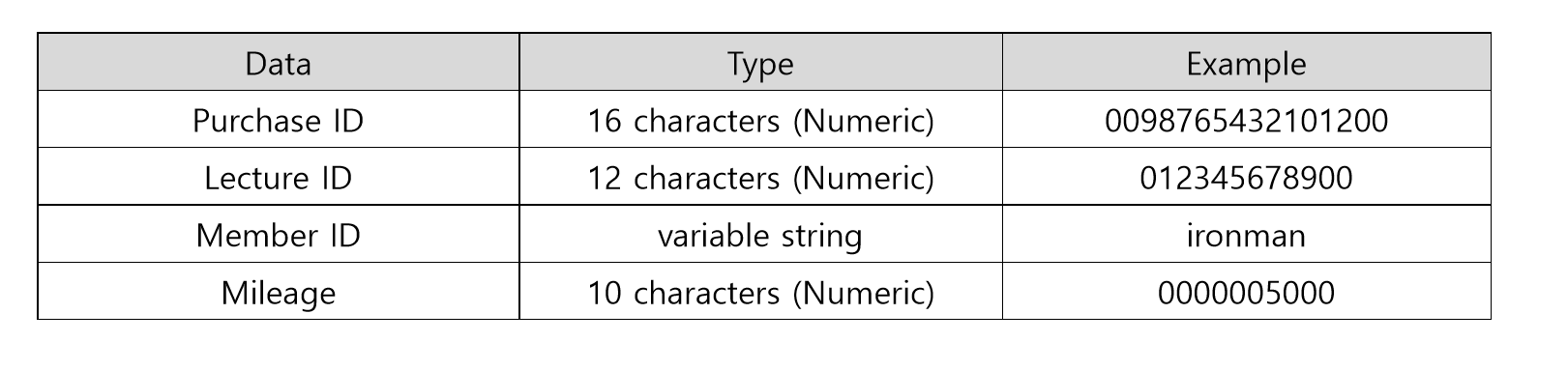
이 객체는 하나의 Lecture에 대한 정보를 나타낸다. Lecture ID, Subject, Level Price, Extension, Due date, Name of teacher, Textbook 등의 private field와 이 객체를 생성할 생성자, field 들을 수정할 method, object 비교/대입 연산에 사용할 오버로드 된 연산자로 구성되어 있다. 각 field 들에 대한 정보는 다음과 같다.



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. **각 파일에 대한 기능 -> 추가된 함수 및 파일**

**main.cpp 추가/변경 사항**

LecturePurchaseSystem(): 관리자 와 일반 사용자 모드의 구분을 위해 log in 기능과 log out 기능을 추가하였다.

proveAssociation(string loging\_id, string want\_ID) : 회원이 데이터를 수정할 때, 권한이 있는지를 검증한다.

SearchingByID(int flag, string filename, string want\_ID): 기존의 data 파일을 활용한 search에서 index 파일을 활용한 search로 변경하였다. Index 파일을 이용한 search는 binary search 이므로 탐색시간이 단축된다.

SearchingByPW(int flag, string filename, string want\_ID, string want\_PW): log in 기능에서 아이디와 비밀번호가 일치하는지 확인한다.

InsertingByID(): index 파일을 이용해 삽입을 하도록 수정했다.

DeletingByID(): index 파일을 이용해 삭제를 하도록 수정했다.

ModifyingByID(): index 파일을 이용해 갱신을 하도록 수정했다.

DeleteMember(string want\_ID): index 파일을 이용해 특정 멤버 레코드를 삭제했다.

DeleteLecture(string want\_ID): index 파일을 이용해 특정 강의 레코드를 삭제했다.

DeletePurchase(string want\_ID): index 파일을 이용해 특정 구매 레코드를 삭제했다.

**member.h 추가/변경 사항**

Level 변수를 추가했다. 이 level 변수를 통해 LecturePurchaseSystem에서 접근 권한을 확인할 수 있다.

**memeber.cpp 추가/변경 사항**

Key() 메소드를 추가했다. 레코드의 Member ID 를 가지고 index 파일에 활용될 key를 생성해주는 함수이다.

**indfile.h 추가/변경 사항**

TextIndexedFile이 구현된 헤더파일이다. Init(RecType & record), Delete(RecType & record) 이 구현되어 있다.

**tindbuff.h 추가/변경 사항**

TextIndexBuffer가 구현된 헤더파일이다.

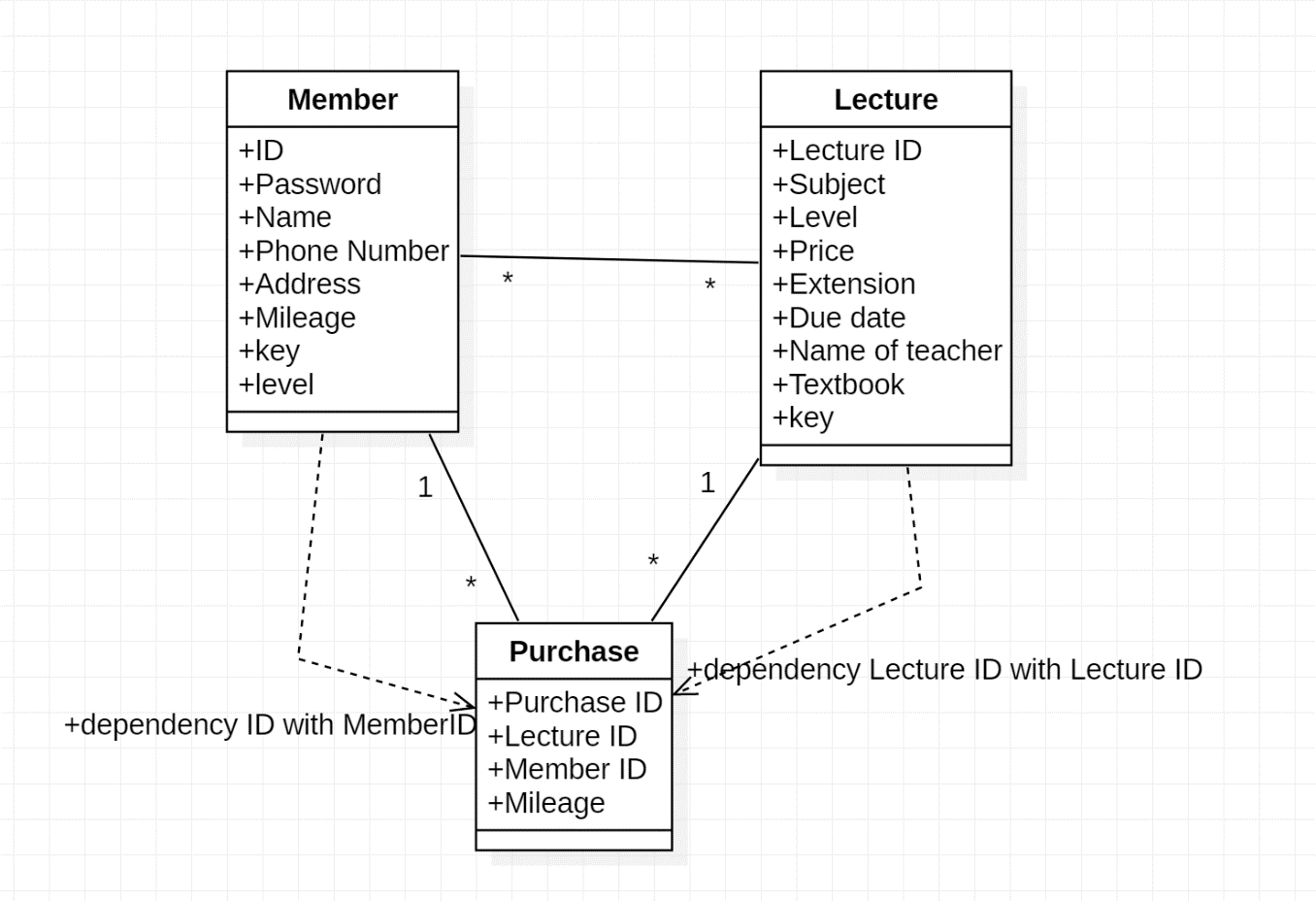
**textind.h 추가/변경 사항**

TextIndex가 구현된 헤더파일이다. FirstRecAddr(const char\* key) . NextRecAddr(int recAddr)이 구현되어 있다.

**textind.cpp 추가/변경 사항**

TextIndex의 하위 메소드 중 기존에 순차접근으로 구현되어 있던, Search 메소드가 Binary Search로 변경되었다.

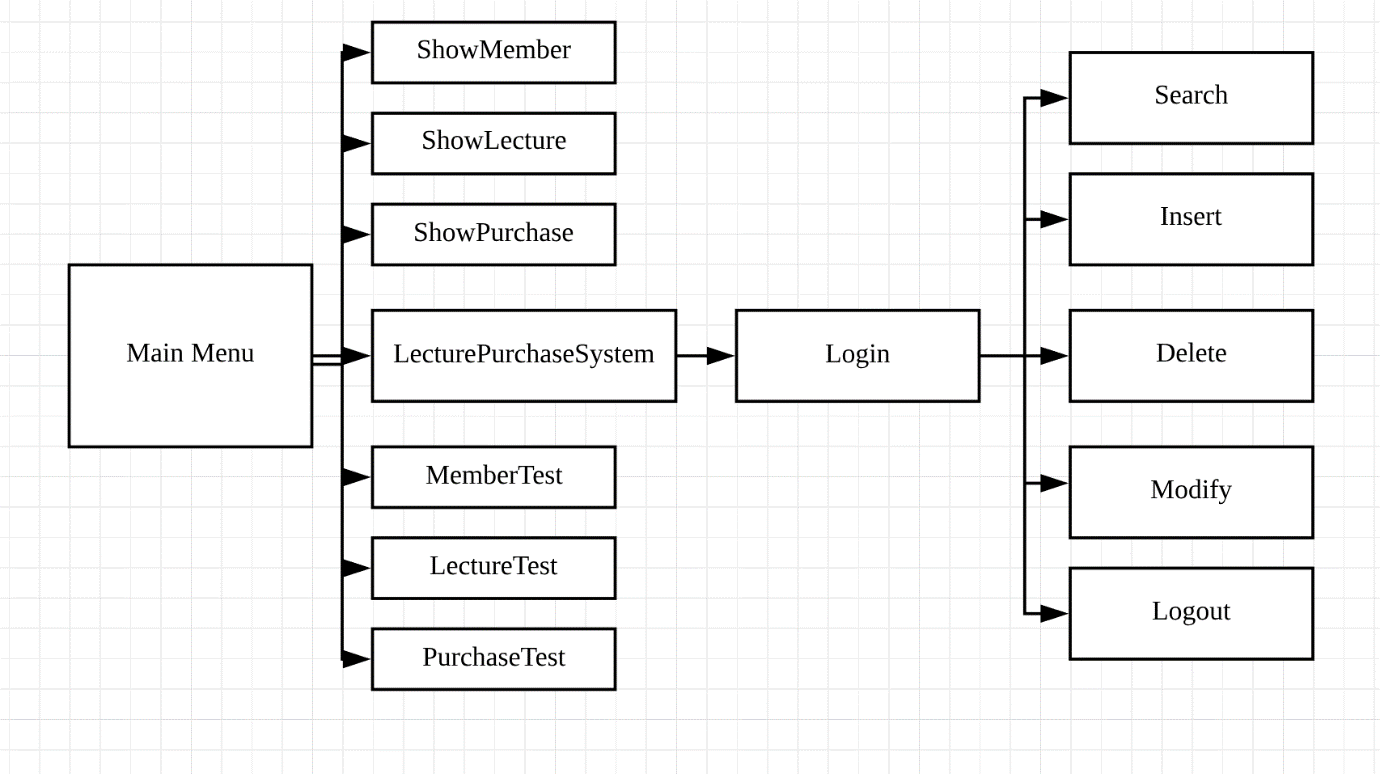
1. **구현한 클래스의 다이어그램(standard UML specification)**



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.

이전의 프로젝트와 달리, Member 클래스에는 key와 level 속성이 추가되었고, Lecture 클래스에는 key 가 추가되었다.

1. **프로그램 흐름도 및 메뉴 이용 방법**



1. **7장 연습 문제 관련**

**13번**

textind.h 파일에 구현되어 있다. FirstRecAddr(const char\* key) 는 가장 작은 키에 대한 참조를 돌려준다. NextRecAddr(int recAddr)는 입력으로 받은 레코드 주소에 대한 다음 참조를 돌려준다.

**16번**

textind.h의 하위 메소드중 기존에 순차접근으로 구현되어 있던 Search method를 Binary Search 로 변경하였다.

**17번**

indfile.h 에 TextIndexedFile이 구현되어 있다. 이 클래스 안에 메소드 중 Delete(RecType & record) 가 구현되어 있다.

**18번**

indfile.h 에 TextIndexedFile이 구현되어 있다. 이 클래스 안에 메소드 중 Init(RecType & record) 가 구현되어 있다.