File Processing

Programming Project #1

20120531

한진욱

1.1 Basic Class

* + 1. 각 클래스의 자료구조

(1) member 클래스

string id;

string password;

패스워드를 저장합니다.

string name;

이름을 저장합니다.

string phoneNumber;

핸드폰번호를 저장합니다.

string address;

주소를 저장합니다.

char mileage[10];

마일리지를 저장합니다.

public:

Member();

Member(const string new\_id);

Member(const Member & s);

위 세 함수는 생성자입니다.

Member & operator = (const Member &s);

bool operator == (const Member &s);

bool operator != (const Member &s);

위 세 함수는 =, == , !=를 재정의 하고 있습니다.

friend istream & operator >> (istream &is, Member &s);

friend ostream & operator << (ostream &os, Member &s);

위 두 함수는 >>, <<를 재정의 하고 있습니다.

void update\_id(const string new\_id) { id = new\_id; }

void update\_name(const string new\_name) { name = new\_name; }

void update\_password(const string new\_number) { password =new\_number; }

void update\_phoneNumber(const string new\_phoneNumber) { phoneNumber = new\_phoneNumber; }

void update\_address(const string new\_address) { address = new\_address; }

void update\_mileage(const char \*new\_mileage) { memcpy(mileage, new\_mileage, 10); }

id, name, password, phonenumber, address, mileage등을 각각 업데이트하는 함수입니다.

bool Pack(IOBuffer & Buffer) const;

file을 write하는데 필요합니다.

bool Unpack(IOBuffer &);

file을 read하는데 필요합니다.

(2) lecture 클래스

char id[12];

string subject;

과목 이름을 저장합니다.

char level[1];

레벨을 저장합니다. ’A’, ‘B’, ‘C’ ,’D’ 중 하나입니다.

char price[7];

가격정보를 저장합니다.

char extension[1];

‘Y’ 아니면 ‘N’을 저장합니다.

char dueDate[2];

‘30’, ‘60’ ‘90’ 중 하나를 저장합니다.

string nameOfTeacher;

교슈의 이름을 저장합니다.

string textBook;

교재의 이름을 저장합니다.

public:

Lecture();

Lecture(const char \* new\_id);

Lecture(const Lecture & s);

위 세 함수는 생성자입니다.

Lecture & operator = (const Lecture &s);

bool operator == (const Lecture &s);

bool operator != (const Lecture &s);

위 세 함수는 =, ==, !=를 각각 재정의합니다.

friend istream & operator >> (istream &is, Lecture &s);

friend ostream & operator << (ostream &os, Lecture &s);

위 두 함수는 >>와 <<를 재정의합니다.

void update\_id(const char \* new\_id) { memcpy(id, new\_id, 12); }

void update\_subject(const string new\_subject) { subject = new\_subject; }

void update\_level(const char\* new\_level) { memcpy(level, new\_level, 1); }

void update\_price(const char \* new\_price) { memcpy(price, new\_price, 7); }

void update\_extension(const char\* new\_extension) { memcpy(extension, new\_extension,1); }

void update\_dueDate(const char\* new\_dueDate) { memcpy(dueDate, new\_dueDate, 2); }

void update\_nameOfTeacher(const string new\_teacher) { nameOfTeacher = new\_teacher; }

void update\_textBook(const string new\_textbook) { textBook = new\_textbook; }

위 함수들은 각각 ID, 과목, 레벨, 가격, 연장여부, 유효기간, 교수이름, 교재이름 등을 새로운 정보로 업데이트합니다.

bool Pack(IOBuffer & Buffer) const;

File을 write하는데 필요합니다.

bool Unpack(IOBuffer &);

file을 read하는데 필요합니다.

(3) Purchase 클래스

char id[16];

char lectureId[12]

강의의 id를 저장합니다.

string memberId;

멤버의 id를 저장합니다.

char mileage[10];

마일리지를 저장합니다.

public:

Purchase();

Purchase(const char \* new\_id);

Purchase(const Purchase & s);

위 함수들은 생성자입니다.

Purchase & operator = (const Purchase &s);

bool operator == (const Purchase &s);

bool operator != (const Purchase &s);

각 함수는 =, != 등을 재정의합니다.

friend istream & operator >> (istream &is, Purchase &s);

friend ostream & operator << (ostream &os, Purchase &s);

위 함수들은 >>, <<를 재정의합니다.

void update\_id(const char \* new\_id) { memcpy(id, new\_id, 16); }

void update\_lectureId(const char \* new\_lectureId) { memcpy(lectureId, new\_lectureId, 12); }

void update\_memberId(const string new\_MemberId) { memberId = new\_MemberId; }

void update\_mileage(const char \* new\_mileage) { memcpy(mileage, new\_mileage, 10); }

각 함수들은 id, lectureId, memberId, mileage등을 새로운 정보로 업데이트합니다.

bool Pack(IOBuffer & Buffer) const;

file을 write하는데 사용합니다.

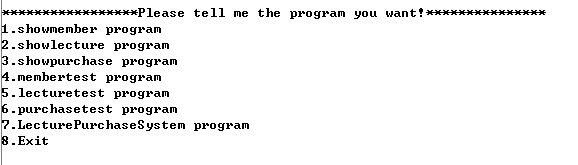
bool Unpack(IOBuffer &);

file을 read하는데 사용합니다.

1.2 Adding method to basic classes

1.2.1 가정

(1) 프로그램을 처음 돌리면 다음과 같이 뜹니다.

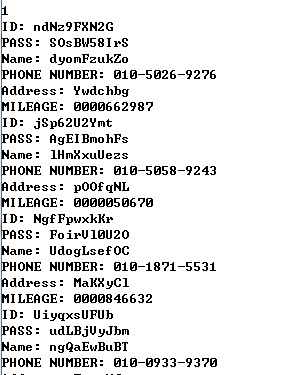


showMember, showLecture, showPurchase 프로그램을 돌리려면 각각 1,2,3을 입력해야 합니다.

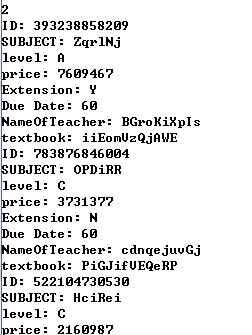
(이 메인 프로그램은 8번을 누르지 않는 이상 무한루프로 돌아갑니다.)

1.2.2 결과

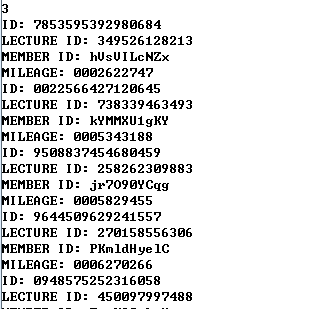
(1) 1을 입력한 경우 member 상위 10명에 대한 정보가 뜹니다.



(2) 2를 입력한 경우 lecture 상위 10개에 대한 정보가 뜹니다.

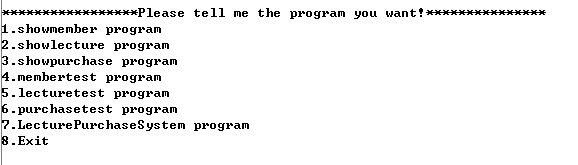


(3) 3을 입력한 경우 purchase에 대한 정보 상위 10가 뜹니다.



1.3 Using IOBuffer

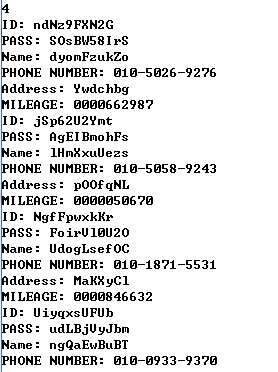
1.3.1 가정



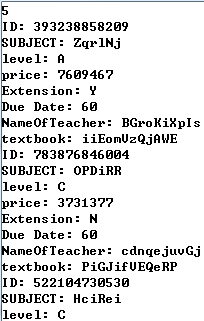
위 메뉴화면에서 membertest, lecturetest, purchasetest 프로그램을 돌리려면 각 4,5,6을 입력해야합니다.

1.3.2 결과

(1) 4를 입력하면 dat 파일에 있는 상위 10명의 member 정보를 보여줍니다.



(2) 5를 입력하면 dat 파일에 있는 상위 10개의 lecture 정보를 보여줍니다.



(3) 6을 입력하면 dat 파일에 있는 상위 10개의 거래 기록을 보여줍니다.

