

[객체지향 프로그래밍]

[클래스 만들기 과제]

과제 체크리스트	확인
1. 이 과제는 내가 직접 연구하고 작성한 것이다.	예
2. 인용한 모든 자료(책, 논문, 보고서, 인터넷 자료 등)의 출처를 밝혔다.	예
3. 정확한 출처 제시나 사용 허락 없이 다른 사람의 아이디어를 가져오지 않았다.	예
4. 이 과제를 다른 사람으로부터 받거나 구매하여 제출하지 않았다.	예
5. 이 과제의 결과물에 나의 학번 이름이 출력되어 있다.	예

■ 담당 교수 : 지정희 교수님

■ 학 번 : 201711430

■ 이 름 : 차준혁

■ 제출일 : 2021년 04월 17일

1 문제정의 및 분석

과제 1에서 수행하였던 영화관 좌석예약 시스템을 클래스화 시키는 과제입니다. 과제 1에서 만든 프로그램의 기능을 똑같이 수행하면서 클래스화 해야 되는데 Movie 클래스와 MovieManager 클래스를 만들고 MovieManager 객체를 생성해서 txt파일에 담긴 정보를 읽고 영화의 개수만큼 Movie 객체를 가리키는 포인터를 담을 수 있는 배열을 동적할당해서 생성합니다. Movie객체의 포인터를 담을 수 있는 배열을 생성했기 때문에 아직 Movie객체가 생성되지 않았으므로 Movie객체도 영화의 개수만큼 동적할당해서 생성합니다. MovieManager 객체에서 메뉴 기능을 구현하고 Movie객체를 선택해서 Movie객체가 예약, 예약취소, 예약확인 기능을 수행하게 만듭니다.

Movie클래스의 필수 멤버로는 1. 영화이름 2. 좌석의 row 3. 좌석의 col 4. 좌석을 동적할당하기 위한 **타입변수 5. 디폴트생성자는 삭제하고 생성자, 복사생성자, 이동생성자 추가 6. 동적할당한 포인터를 따로 해제하기 위해서 소멸자가 있어야 합니다.

MovieManager클래스의 필수 멤버로는 1. Movie객체를 동적 할당해서 생성하기 위한 Movie** movie 2. 디폴트 생성자는 삭제하고 생성자, 복사생성자, 이동생성자 추가 3. 동적할당한 포인터를 따로 해제하기 위해서 소멸자 4. 메뉴 구동하는 멤버 함수startMenu()가 있어야합니다.

2 주요 소스코드 설명

Movie 클래스

Private:

int movieselect: 영화의 순서 변수

string MovieName: 영화의 이름 변수

int row, col: 영화의 행과 열 변수

int availableNum: 영화의 이용 가능한 좌석 수 변수

bool** seats: 2차원 배열로 생성할 좌석의 예약 유무를 확인하기 위한 변수

Public:

Movie() = delete: 디폴트 생성자 삭제

Movie(const int movieselect, const string MovieName, const int row, const int col):

생성자

~Movie(): 소멸자

Movie(Movie& movie): 복사생성자

Movie(Movie&& movie)noexcept: 이동생성자

void loadSeats(): 좌석 배열의 정보 동적 할당 및 초기화하는 함수

void showSeats(): 좌석 정보 출력하는 함수

void showMovieData(): Movie객체의 변수 정보 출력하는 함수

void reserveSeats(): 좌석 예약 함수

void cancelSeats(char c2, char c3): 좌석 취소 함수

void checkSeats(char c2, char c3): 좌석 예약 확인 함수

```
Movie::Movie(const int movieselect, const string mname, const int Row, const int Col):movieselect(movieselect),MovieName(mname), row(Row), col(Col), availableNum(Row*Col) {
    loadSeats();
    //cout << "Movie 생성자 호출" << endl;
}
```

```
void Movie::loadSeats()
{
    this->seats = new bool* [this->row];
    for (int i = 0; i < this->row; i++) {
        this->seats[i] = new bool[this->col];
    }

    for (int i = 0; i < this->row; i++) {
        for (int j = 0; j <this->col; j++) {
            this->seats[i][j] = false;
        }
    }
}
```

Movie객체 생성과 동시에 영화의 순서, 영화 이름, 영화좌석의 행, 열, 이용가능한 좌석 수를 초기화하고 loadSeats()함수로 좌석배열 동적 할당 및 초기화합니다.

```
Movie::~~Movie()
{
    cout << "Movie 소멸자" << endl;
    if (seats != nullptr) {
        for (int i = 0; i < this->row; i++) {
            delete[] seats[i];
        }
        delete[] seats;
    }
}
```

```

    }
    seats = nullptr;
}

```

Movie객체 소멸자가 생성될 때 동적할당한 좌석의 배열을 delete합니다.

```

Movie::Movie(Movie& movie) :movieselect(movie.movieselect), MovieName(movie.MovieName),
row(movie.row), col(movie.col), availableNum(movie.availableNum)
{
    loadSeats();
    for (int i = 0; i < this->row; i++) {
        for (int j = 0; j < this->col; j++) {
            this->seats[i][j] = movie.seats[i][j];
        }
    }
    cout << "Movie 복사생성자" << endl;
}

```

Movie객체 movie의 변수와 좌석 정보를 복사생성자의 변수와 좌석정보를 대입하여 초기화합니다.

loadSeats()를 이용하여 복사생성자의 좌석 배열을 따로 동적 할당해서 깊은 복사를 실행합니다.

```

Movie::Movie(Movie&& movie)noexcept:movieselect(movie.movieselect),
MovieName(movie.MovieName), row(movie.row), col(movie.col),
availableNum(movie.availableNum)
{
    cout << "Movie 이동생성자" << endl;
    this->seats = movie.seats;
    movie.seats = nullptr;
}

```

Movie객체 movie의 변수와 좌석 정보를 이동생성자의 변수와 좌석정보에 대입하여 초기화합니다. 나중에 소멸자의 의한 중복 delete를 막기 위해서 Movie객체의 movie변수의 seats는 nullptr로 초기화 해줍니다.

```

void Movie::showSeats()
{
    cout << endl;
    for (int i = 0; i < col; i++)
        cout << 'Wt' << i + 1;
    cout << "WnWn";
    for (int i = 0; i < row; i++) {
        cout << static_cast<char>('A' + i) << 'Wt';
        for (int j = 0; j < col; j++) {
            cout << (seats[i][j] ? "●" : "○") << 'Wt';

```

```

    }
    cout << endl;
}
}

```

Movie객체의 좌석의 정보를 출력하는 함수입니다.

```

void Movie::showMovieData()
{
    string name = this->MovieName;
    int namelen = (38 - name.length());
    cout << (this->movieselect + 1) << " ) " << this->MovieName;
    if (this->availableNum == 0) {
        cout << setfill(' ') << setw(namelen + 3);
        cout << "매진" << endl;
    }
    else {
        cout << setfill(' ') << setw(namelen);
        cout << this->availableNum << "/" << (this->row) * (this->col) << endl;
    }
}

```

Movie객체의 영화 이름, 남은 좌석 수, 전체 좌석수와 같은 영화 정보를 출력하는 함수입니다.

```

void Movie::reserveSeats()
{
    cout << "\n예매할 좌석을 선택하세요(ex) A 1, B 2) : ";
    char x, y;
    cin >> x >> y;
    char MDRow, MDCol; //영화관 좌석의 행과 열을 뜻함
    MDRow = static_cast<char>('A' + (this->row) - 1);
    MDCol = static_cast<char>('1' + (this->col) - 1);
    if (x < 'A' || x > MDRow) {
        cout << "\n죄송하지만 존재하지 않는 좌석입니다." << endl;
    }

    else if (y < '1' || y > MDCol) {
        cout << "\n죄송하지만 존재하지 않는 좌석입니다." << endl;
    }

    else if (seats[x - 'A'][y - '1']) {
        cout << "\n이미 예약된 자리입니다." << endl;
    }
    else {
        seats[x - 'A'][y - '1'] = true;
        (this->availableNum)--;
        cout << "\n예약이 완료되었습니다." << endl;
    }
}

```

```

        cout << "예약번호는 " << static_cast<char>(this->movieselect + 'A');
        cout << static_cast<char>(x - 'A' + 'C');
        cout << static_cast<char>(y - '1' + 'D') << " 입니다." << endl;
    }
}

```

Movie객체의 영화 예매를 실행하고 예약번호를 출력하는 함수입니다.

```

void Movie::cancelSeats(char c2, char c3)
{
    int dNum2 = static_cast<int>(c2 - 'C');
    int dNum3 = static_cast<int>(c3 - 'D');
    if (dNum2 < 0 || dNum2 > (this->row) - 1 || dNum3 < 0 || dNum3 > (this->col) - 1) {
        cout << "예약번호가 존재하지 않습니다." << endl;
    }
    else {
        if (seats[dNum2][dNum3]) {
            seats[dNum2][dNum3] = 0;
            this->availableNum++;
            cout << "예약이 취소되었습니다." << endl;
        }
        else {
            cout << "예약번호가 존재하지 않습니다." << endl;
        }
    }
}

```

Movie객체의 예매한 영화의 예약번호를 입력 받고 예약이 존재한다면 취소를 실행하는 함수입니다.

```

void Movie::checkSeats(char c2, char c3)
{
    int searchNum2 = static_cast<int>(c2 - 'C');
    int searchNum3 = static_cast<int>(c3 - 'D');
    if (searchNum2 < 0 || searchNum2 > (this->row) - 1 || searchNum3 < 0 || searchNum3 > (this->col) - 1) {
        cout << "예약번호가 존재하지 않습니다." << endl;
    }
    else {
        if (seats[searchNum2][searchNum3]) {
            cout << "예매하신 영화 : " << this->MovieName << endl;
            cout << "예매하신 좌석 : " << static_cast<char>(searchNum2 + 'A')
            << (searchNum3 + 1) << endl;
            cout << "=====영화 예매 좌석=====\\n";
            cout << endl;
            for (int i = 0; i < this->col; i++)
                cout << 'A' + i << " ";
        }
    }
}

```

```

        cout << "WnWn";
        for (int i = 0; i < this->row; i++) {
            cout << static_cast<char>('A' + i) << 'Wt';
            for (int j = 0; j < this->col; j++) {
                cout << (((i == searchNum2) && (j == searchNum3)) ?
"☆" : "○") << 'Wt';
            }
            cout << endl;
        }
        system("pause");
    }
    else {
        cout << "Wn예약번호가 존재하지 않습니다." << endl;
    }
}
}

```

Movie객체의 예매한 영화의 예약번호를 입력 받고 예약이 존재한다면 해당 영화 정보와 좌석 정보의 위치를 출력하는 함수입니다.

MovieManager 클래스

Private:

Movie** movie: Movie객체를 가리키는 포인터를 담은 배열의 이름 변수
int menuNum: 실행할 메뉴 번호를 입력 받을 변수
int MovieNum: txt파일에서 읽을 영화의 개수 변수
string fileloc: 읽을 txt파일 경로명 변수

Public:

MovieManager() = delete: 디폴트 생성자 삭제

MovieManager(string fileloc): 생성자

~MovieManager(): 소멸자

MovieManager(const MovieManager& moviemanager): 복사생성자

MovieManager(MovieManager&& moviemanager)noexcept: 이동생성자

void startMenu(): 메뉴번호를 입력받고 선택한 메뉴를 출력하고 기능을 실행하는 함수

void reserveMovie(): 영화의 리스트를 출력, 예매할 영화의 번호를 입력 받고 그 영화객체의 예약함수를 실행

void cancelMovie(): 예매취소 할 영화의 예약번호를 입력 받고 그 예약 번호의 예약이

존재한다면 그 영화 객체의 예약취소 함수를 실행하는 함수

void checkMovie(): 예매확인 할 영화의 예약번호를 입력 받고 그 예약 번호의 예약이 존재한다면 그 영화 객체의 예약확인 함수를 실행하는 함수

```
MovieManager::MovieManager(string fileloc):fileloc(fileloc)
{
    this->menuNum = -1;
    this->MovieNum = -1;
    ifstream fin(fileloc);
    if (!fin.is_open()) {
        cerr << "파일 오픈 실패";
        exit(0);
    }
    fin >> (this->MovieNum);
    if ((this->MovieNum) > 0) {
        this->movie = new Movie * [this->MovieNum];
        while (!fin.eof()) {
            for (int i = 0; i < (this->MovieNum); i++) {
                string mname;
                int row, col;
                fin >> mname;
                fin >> row;
                fin >> col;
                movie[i] = new Movie(i, mname, row, col); //Movie객체 동적
                //cout << "Movie객체 동적 생성" << endl;
            }
        }
    }
    else {
        movie = nullptr;
    }
    //cout << "MovieManager 생성자" << endl;
}
```

MovieManager객체 생성과 동시에 fileloc변수의 txt파일을 읽어서 영화의 개수를 읽고 영화의 개수만큼 영화객체를 가리키는 포인터를 원소로 가지는 배열을 동적 할당하여 생성합니다. 또 영화의 개수만큼 영화의 객체를 동적 할당하여 생성해서 배열에 대입합니다.

```
MovieManager::~MovieManager()
{
    cout << "MovieManager 소멸자" << endl;
}
```



```

        if (movie != nullptr) {
            for (int i = 0; i < (this->MovieNum); i++) {
                delete movie[i];
                //cout << "delete Movie객체" << endl;
            }
            delete[] movie;
            //cout << "delete Movie객체의 포인터 배열" << endl;
        }
        movie = nullptr;
    }
}

```

Movie객체 소멸자가 생성될 때 동적할당한 객체와 객체의 포인터를 원소로 가지는 배열을 delete합니다.

```

MovieManager::MovieManager(const MovieManager& moviemanager)
    :fileloc(moviemanager.fileloc),MovieNum(moviemanager.MovieNum),
    menuNum(moviemanager.menuNum)
{
    cout << "MovieManager 복사생성자" << endl;
    if ((moviemanager.MovieNum) > 0) {
        this->movie = new Movie * [moviemanager.MovieNum];
        for (int i = 0; i < (moviemanager.MovieNum); i++) {
            movie[i] = new Movie(*(moviemanager.movie[i])); //Movie객체
            //cout << "Movie객체 동적 생성" << endl;
        }
    }
    else {
        movie = nullptr;
    }
}
}

```

MovieManager복사생성자를 생성할 때 Movie객체와 Movie객체의 포인터를 담는 배열을 따로 동적 할당하여 생성하고 깊은 복사를 실행합니다.

```

MovieManager::MovieManager(MovieManager&& moviemanager) noexcept
    :fileloc(moviemanager.fileloc),MovieNum(moviemanager.MovieNum),
    menuNum(moviemanager.menuNum)
{
    cout << "MovieManager 이동생성자" << endl;
    this->movie = moviemanager.movie;
    moviemanager.movie = nullptr;
}
}

```

MovieManager이동생성자를 생성할 때 moviemanager.movie를 이동생성자의 movie에 대입해서 초기화하고 중복 delete를 막기위해 moviemanager.movie를 nullptr로 초기화해줍니다.

```

void MovieManager::startMenu()
{
    if (this->movie != nullptr) {
        do {
            CJH::printName();
            cout << "=====차준혁 시네마===== " << endl;
            cout << "1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료" << endl;
            cout << "Wn메뉴를 선택하세요 : ";
            cin >> this->menuNum;
            switch (this->menuNum) {
                case 1:
                    reserveMovie();
                    break;
                case 2:
                    cancelMovie();
                    break;
                case 3:
                    checkMovie();
                    break;

                case 4:
                    cout << "Wn예매 프로그램을 종료합니다. 안녕히 가세요
^.^Wn";

                    break;
                default:
                    cout << "잘못 입력하셨습니다.Wn";
                    break;
            }
            cout << "Wn";
        } while (this->menuNum != 4);
    }
}

```

메뉴를 출력하고 메뉴번호를 입력받아서 해당하는 메뉴의 기능을 실행하는 함수입니다.

```

void MovieManager::reserveMovie()
{
    int select;
    cout << "Wn=====영화 예매=====Wn";
    for (int i = 0; i < MovieNum; i++) {
        movie[i]->showMovieData();
    }
    cout << endl << "예매할 영화를 선택하세요 : ";
    cin >> select;
    movie[select - 1]->showSeats();
    movie[select - 1]->reserveSeats();
}

```

예매할 영화의 번호를 입력 받아서 해당 영화 객체의 좌석정보출력함수와 예약함수를 실행하는 함수입니다.

```
void MovieManager::cancelMovie()
{
    char c1, c2, c3;
    cout << "취소할 영화의 예약번호를 입력해주세요(ex) 예약번호가 AEF라면 입력: AEF): ";
    cin >> c1 >> c2 >> c3;
    int dNum1 = static_cast<int>(c1 - 'A');
    if (dNum1 < 0 || dNum1 > (this->MovieNum)-1) {
        cout << "예약번호가 존재하지 않습니다." << endl;
    }
    else {
        movie[dNum1]->cancelSeats(c2, c3);
    }
}
```

예매취소 할 영화의 예약번호를 입력 받아서 예약번호가 존재한다면 해당 영화 객체의예약취소 함수를 실행하는 함수입니다.

```
void MovieManager::checkMovie()
{
    char c1, c2, c3;
    cout << "예약 확인=====" << endl;
    cout << "예약번호를 입력하세요(ex) 예약번호가 AEF라면 입력: AEF): ";
    cin >> c1 >> c2 >> c3;
    int searchNum1 = static_cast<int>(c1 - 'A');
    if (searchNum1 < 0 || searchNum1 > (this->MovieNum) - 1) {
        cout << "예약번호가 존재하지 않습니다." << endl;
    }
    else {
        movie[searchNum1]->checkSeats(c2, c3);
    }
}
```

예매확인 할 영화의 예약번호를 입력 받아서 예약번호가 존재한다면 해당 영화 객체의 예약확인함수를 실행하는 함수입니다.

```
#include <iostream>
#include "MovieManager.h"
using namespace std;

//메모리 누수 체크!
#ifdef _DEBUG
#ifdef DBG_NEW
#define DBG_NEW new ( _NORMAL_BLOCK , __FILE__ , __LINE__ )
#define new DBG_NEW
```

```

#endif // !DBG_NEW
#endif

MovieManager returnManager(string fileloc) {
    MovieManager temp(fileloc);
    return temp;
}
Movie returnMovie() {
    Movie temp(5, "라라랜드", 5, 3);
    return temp;
}

int main() {
    //메모리 누수 체크!
    _CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF);

    cout << "=====생성자======" << endl<<endl;
    MovieManager June("moviedata.txt");
    June.startMenu();

    cout << "=====복사생성자======" << endl<<endl;
    MovieManager Hyeok(June);
    Hyeok.startMenu();

    cout << "=====이동생성자======" << endl << endl;
    MovieManager Cha = returnManager("moviedata.txt");
    Cha.startMenu();

    //Movie객체 이동생성자
    /*cout << "201711430 컴퓨터공학부 차준혁Wn" << endl << endl;
    Movie testmovie = returnMovie();
    testmovie.showMovieData();
    testmovie.showSeats();
    */

    /*MovieManager Cha(move(June));
    Cha.startMenu();
    June.startMenu();*/
}

```

객체를 생성하고 객체의 기능을 실행, test하는 프로그램의 main함수입니다.

```
학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====
1) 소울 15/15
2) 어벤져스:엔드게임 30/30
3) 너의모든것 25/25
4) 청설 12/12

예매할 영화를 선택하세요 : 4

      1      2      3
A      ○      ○      ○
B      ○      ○      ○
C      ○      ○      ○
D      ○      ○      ○

예매할 좌석을 선택하세요(ex) A 1, B 2) : A1

예약이 완료되었습니다.
예약번호는 DCD 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====
1) 소울 15/15
2) 어벤져스:엔드게임 30/30
3) 너의모든것 25/25
4) 청설 11/12

예매할 영화를 선택하세요 : 4

      1      2      3
A      ●      ○      ○
B      ○      ○      ○
C      ○      ○      ○
D      ○      ○      ○

예매할 좌석을 선택하세요(ex) A 1, B 2) : A1

이미 예약된 자리입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====
1) 소울 15/15
2) 어벤져스:엔드게임 30/30
```

C:\> 선택 C:\Users\차준혁\source\repos\Jtheater2\Debug\Jtheater2.exe

```
B      ○      ○      ○
C      ○      ○      ○
D      ○      ○      ○
```

예매할 좌석을 선택하세요(ex) A 1, B 2) : A1

이미 예약된 자리입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====

1) 소울	15/15
2) 어벤져스:엔드게임	30/30
3) 너의모든것	25/25
4) 청설	11/12

예매할 영화를 선택하세요 : 4

	1	2	3
A	●	○	○
B	○	○	○
C	○	○	○
D	○	○	○

예매할 좌석을 선택하세요(ex) A 1, B 2) : A2

예약이 완료되었습니다.

예약번호는 DCE 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====

1) 소울	15/15
2) 어벤져스:엔드게임	30/30
3) 너의모든것	25/25
4) 청설	10/12

예매할 영화를 선택하세요 : 4

	1	2	3
A	●	●	○
B	○	○	○
C	○	○	○
D	○	○	○

예매할 좌석을 선택하세요(ex) A 1, B 2) : A3

예약이 완료되었습니다.

예약번호는 DCF 입니다.

학번 : 201711430 이름 : 차준혁

cs 선택 C:\Users\차준혁\source\repos\Jtheater2\Debug\Jtheater2.exe

예매할 좌석을 선택하세요(ex) A 1, B 2) : A3

예약이 완료되었습니다.
예약번호는 DCF 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 3

=====예약 확인=====

예약번호를 입력하세요(ex) 예약번호가 AEF라면 입력: AEF): DCF

예매하신 영화 : 청설

예매하신 좌석 : A3

=====영화 예매 좌석=====

	1	2	3
A	○	○	☆
B	○	○	○
C	○	○	○
D	○	○	○

계속하려면 아무 키나 누르십시오 . . .

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 2

취소할 영화의 예약번호를 입력해주세요(ex) 예약번호가 AEF라면 입력: AEF): DCF

예매가 취소되었습니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====

1) 소울	15/15
2) 어벤져스:엔드게임	30/30
3) 너의모든것	25/25
4) 청설	10/12

예매할 영화를 선택하세요 : 4

	1	2	3
A	●	●	○
B	○	○	○
C	○	○	○
D	○	○	○

예매할 좌석을 선택하세요(ex) A 1, B 2) : A3

예약이 완료되었습니다.
예약번호는 DCF 입니다.

cs> 선택 C:\Users\#준혁#source\repos\Jtheater2\Debug\Jtheater2.exe

예약이 완료되었습니다.
예약번호는 DCF 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====

1) 소울	15/15
2) 어벤져스:엔드게임	30/30
3) 너의모든것	25/25
4) 청설	9/12

예매할 영화를 선택하세요 : 4

	1	2	3
A	●	●	●
B	○	○	○
C	○	○	○
D	○	○	○

예매할 좌석을 선택하세요(ex) A 1, B 2) : B1

예약이 완료되었습니다.
예약번호는 DDD 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====

1) 소울	15/15
2) 어벤져스:엔드게임	30/30
3) 너의모든것	25/25
4) 청설	8/12

예매할 영화를 선택하세요 : 4

	1	2	3
A	●	●	●
B	●	○	○
C	○	○	○
D	○	○	○

예매할 좌석을 선택하세요(ex) A 1, B 2) : B2

예약이 완료되었습니다.
예약번호는 DDE 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

cs 선택 C:\Users\#준혁\source\repos\Jtheater2\Debug\Jtheater2.exe

예약이 완료되었습니다.
예약번호는 DDE 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====

1) 소울	15/15
2) 어벤저스:엔드게임	30/30
3) 너의모든것	25/25
4) 청설	7/12

예매할 영화를 선택하세요 : 4

	1	2	3
A	●	●	●
B	●	●	○
C	○	○	○
D	○	○	○

예매할 좌석을 선택하세요(ex) A 1, B 2) : B3

예약이 완료되었습니다.
예약번호는 DDF 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====

1) 소울	15/15
2) 어벤저스:엔드게임	30/30
3) 너의모든것	25/25
4) 청설	6/12

예매할 영화를 선택하세요 : 4

	1	2	3
A	●	●	●
B	●	●	●
C	○	○	○
D	○	○	○

예매할 좌석을 선택하세요(ex) A 1, B 2) : C1

예약이 완료되었습니다.
예약번호는 DED 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

```

4) 청설                                     5/12

예매할 영화를 선택하세요 : 4

      1      2      3
A      ●      ●      ●
B      ●      ●      ●
C      ●      ○      ○
D      ○      ○      ○

예매할 좌석을 선택하세요(ex) A 1, B 2) : 02

예약이 완료되었습니다.
예약번호는 DEE 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====
1) 소울                                     15/15
2) 어벤저스:엔드게임                     30/30
3) 너의모든것                             25/25
4) 청설                                    4/12

예매할 영화를 선택하세요 : 4

      1      2      3
A      ●      ●      ●
B      ●      ●      ●
C      ●      ○      ○
D      ○      ○      ○

예매할 좌석을 선택하세요(ex) A 1, B 2) : 03

예약이 완료되었습니다.
예약번호는 DEF 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====
1) 소울                                     15/15
2) 어벤저스:엔드게임                     30/30
3) 너의모든것                             25/25
4) 청설                                    3/12

예매할 영화를 선택하세요 : 4

      1      2      3
A      ●      ●      ●
B      ●      ●      ●
C      ●      ●      ●

```

cs 선택 C:\Users\차준혁\source\repos\Jtheater2\Debug\Jtheater2.exe

```

      1      2      3
A      ●      ●      ●
B      ●      ●      ●
C      ●      ●      ●
D      ○      ○      ○

예매할 좌석을 선택하세요(ex) A 1, B 2) : D1

예약이 완료되었습니다.
예약번호는 DFD 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====
1) 소울                      15/15
2) 어벤져스:엔드게임        30/30
3) 너의모든것                25/25
4) 청설                      2/12

예매할 영화를 선택하세요 : 4

      1      2      3
A      ●      ●      ●
B      ●      ●      ●
C      ●      ●      ●
D      ●      ○      ○

예매할 좌석을 선택하세요(ex) A 1, B 2) : D2

예약이 완료되었습니다.
예약번호는 DFE 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====
1) 소울                      15/15
2) 어벤져스:엔드게임        30/30
3) 너의모든것                25/25
4) 청설                      1/12

예매할 영화를 선택하세요 : 4

      1      2      3
A      ●      ●      ●
B      ●      ●      ●
C      ●      ●      ●
D      ●      ●      ○

예매할 좌석을 선택하세요(ex) A 1, B 2) : D3
```

선택 C:\Users\차준혁\source\repos\Jtheater2\Debug\Jtheater2.exe

```
D
예매할 좌석을 선택하세요(ex) A 1, B 2) : D3
예약이 완료되었습니다.
예약번호는 DFF 입니다.
학번 : 201711430 이름 : 차준혁
=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료
메뉴를 선택하세요 : 1
=====영화 예매=====
1) 소울 15/15
2) 어벤저스:엔드게임 30/30
3) 너의모든것 25/25
4) 청설매진
예매할 영화를 선택하세요 : 4
      1      2      3
A      ●      ●      ●
B      ●      ●      ●
C      ●      ●      ●
D      ●      ●      ●
예매할 좌석을 선택하세요(ex) A 1, B 2) : D3
이미 예약된 자리입니다.
학번 : 201711430 이름 : 차준혁
=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료
메뉴를 선택하세요 : 2
취소할 영화의 예약번호를 입력해주세요(ex) 예약번호가 AEF라면 입력: AEF): DFG
예약번호가 존재하지 않습니다.
학번 : 201711430 이름 : 차준혁
=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료
메뉴를 선택하세요 : 2
취소할 영화의 예약번호를 입력해주세요(ex) 예약번호가 AEF라면 입력: AEF): DFF
예매가 취소되었습니다.
학번 : 201711430 이름 : 차준혁
=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료
메뉴를 선택하세요 : 1
=====영화 예매=====
1) 소울 15/15
2) 어벤저스:엔드게임 30/30
```

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 2

취소할 영화의 예약번호를 입력해주세요(ex) 예약번호가 AEF라면 입력: AEF): DFG

예약번호가 존재하지 않습니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 2

취소할 영화의 예약번호를 입력해주세요(ex) 예약번호가 AEF라면 입력: AEF): DFF

예매가 취소되었습니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====

1) 소울	15/15
2) 어벤져스:엔드게임	30/30
3) 너의모든것	25/25
4) 청설	1/12

예매할 영화를 선택하세요 : 4

	1	2	3
A	●	●	●
B	●	●	●
C	●	●	●
D	●	●	○

예매할 좌석을 선택하세요(ex) A 1, B 2) : D3

예약이 완료되었습니다.

예약번호는 DFF 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====

1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 4

예매 프로그램을 종료합니다. 안녕히 가세요 ^.^

C:\Users\차준혁\source\repos\Jtheater2\Debug\Jtheater2.exe(프로세스 25224개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

<생성자와 소멸자 테스트>

1. 생성자, 소멸자

```
MovieManager June("moviedata.txt");  
June.startMenu();
```

MovieManager 객체 June을 인자값 "moviedata.txt"을 주고 생성합니다.

```
Microsoft Visual Studio 디버그 콘솔  
MovieManager 생성자  
Movie 생성자  
Movie 생성자  
Movie 생성자  
Movie 생성자  
학번 : 201711430 이름 : 차준혁  
  
=====차준혁 시네마=====  
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료  
메뉴를 선택하세요 : 1  
  
=====영화 예매=====  
1) 소울 15/15  
2) 어벤저스:엔드게임 30/30  
3) 너의모든것 25/25  
4) 청설 12/12  
예매할 영화를 선택하세요 : 4  
  
1 2 3  
A ○ ○ ○  
B ○ ○ ○  
C ○ ○ ○  
D ○ ○ ○  
예매할 좌석을 선택하세요(ex) A 1, B 2) : A1  
예약이 완료되었습니다.  
예약번호는 D0D 입니다.  
학번 : 201711430 이름 : 차준혁  
  
=====차준혁 시네마=====  
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료  
메뉴를 선택하세요 : 4  
예매 프로그램을 종료합니다. 안녕히 가세요 ^.^  
  
MovieManager 소멸자  
Movie 소멸자  
Movie 소멸자  
Movie 소멸자  
Movie 소멸자  
C:\Users\차준혁\source\repos\Jtheater2\Debug\Jtheater2.exe(프로세스 21360개)이(가) 종료되었습니다(코드: 0개).  
이 창을 닫으려면 아무 키나 누르세요...
```

위의 결과와 같이 MovieManager객체의 생성자가 먼저 만들어지고 만들어진 MovieManager객체가 Movie객체를 동적 할당하여 생성합니다.

메뉴 종료를 선택하면 MovieManager객체의 소멸자가 생성됩니다. 원래 객체의 소멸 순서는 생성된 순서의 역순으로 소멸하지만 Movie객체를 동적 할당하여 생성했기 때문에 MovieManager객체의 소멸자가 delete연산자를 실행하여 객체를 직접 소멸시켜야 합니다. 그래서 MovieManager객체의 소멸자가 먼저 생성되고 그 소멸자가 Movie객체를 delete하면서 Movie객체의 소멸자가 생성되고 객체가 소멸되는 것입니다.

2.복사 생성자

```
MovieManager June( "moviedata.txt" );  
MovieManager Hyeok(June);  
Hyeok.startMenu();
```

MovieManager객체 June을 생성하고 MovieManager객체 Hyeok을 복사 생성합니다.

Microsoft Visual Studio 디버그 콘솔

```
MovieManager 생성자
Movie 생성자
Movie 생성자
Movie 생성자
Movie 생성자
MovieManager 복사생성자
Movie 복사생성자
Movie 복사생성자
Movie 복사생성자
Movie 복사생성자
학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 1

=====영화 예매=====
1) 소울 15/15
2) 어벤져스:엔드게임 30/30
3) 너의모든것 25/25
4) 청설 12/12

예매할 영화를 선택하세요 : 1

      1      2      3
A      ○      ○      ○
B      ○      ○      ○
C      ○      ○      ○
D      ○      ○      ○
E      ○      ○      ○

예매할 좌석을 선택하세요(ex) A 1, B 2) : A1

예약이 완료되었습니다.
예약번호는 ACD 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 3

=====예약 확인=====
예약번호를 입력하세요(ex) 예약번호가 AEF라면 입력: AEF): ACD
예매하신 영화 : 소울
예매하신 좌석 : A1

=====영화 예매 좌석=====

      1      2      3
A      ☆      ○      ○
B      ○      ○      ○
C      ○      ○      ○
D      ○      ○      ○
E      ○      ○      ○
계속하려면 아무 키나 누르십시오 . . .

학번 : 201711430 이름 : 차준혁
```

먼저 객체 June을 생성했기 때문에 MovieManager생성자가 생성되고 Movie 생성자를 생성합니다. 그리고 객체 Hyeok을 복사 생성해서 MovieManager 복사 생성자가 생성되고 Movie객체를 또 복사 생성해서 Movie복사 생성자가 생성됩니다.


```
Microsoft Visual Studio 디버그 콘솔

      1      2      3
A      ○      ○      ○
B      ○      ○      ○
C      ○      ○      ○
D      ○      ○      ○
E      ○      ○      ○

예매할 좌석을 선택하세요(ex) A 1, B 2) : A1

예약이 완료되었습니다.
예약번호는 ACD 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 3

=====예약 확인=====
예약번호를 입력하세요(ex) 예약번호가 AEF라면 입력: AEF): ACD
예매하신 영화 : 소울
예매하신 좌석 : A1

=====영화 예매 좌석=====

      1      2      3
A      ☆      ○      ○
B      ○      ○      ○
C      ○      ○      ○
D      ○      ○      ○
E      ○      ○      ○
계속하려면 아무 키나 누르십시오 . . .

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 4

예매 프로그램을 종료합니다. 안녕히 가세요 ^.^

MovieManager 소멸자
Movie 소멸자
Movie 소멸자
Movie 소멸자
Movie 소멸자
MovieManager 소멸자
Movie 소멸자
Movie 소멸자
Movie 소멸자
Movie 소멸자
```

메뉴 종료를 선택하면 MovieManager객체의 소멸자가 생성됩니다. MovieManager객체 Hyeok의 MovieManager 소멸자가 먼저 생성되고 Movie객체를 delete해서 Movie소멸자가 생성됩니다. 그리고 MovieManager객체 June의 MovieManager 소멸자가 생성되고 Movie 객체를 delete해서 Movie소멸자가 생성됩니다.

3.이동생성자

```

MovieManager returnManager(string fileloc) {
    MovieManager temp(fileloc);
    return temp;
}

```

MovieManager객체의 이동생성자 생성을 위해서 만든 MovieManager를 반환하는 함수입니다.

```

cout << "=====이동생성자===== " << endl << endl;
MovieManager Cha = returnManager("moviedata.txt");
Cha.startMenu();

```

MovieManager객체 Cha는 returnManager함수가 반환하는 MovieManager 객체를 받아서 생성됩니다.

```

Microsoft Visual Studio 디버그 콘솔
=====이동생성자=====
MovieManager 생성자
Movie 생성자
Movie 생성자
Movie 생성자
Movie 생성자
MovieManager 이동생성자
MovieManager 소멸자
학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료
메뉴를 선택하세요 : 1

=====영화 예매=====
1) 소울 15/15
2) 어벤져스:엔드게임 30/30
3) 너의모든것 25/25
4) 청설 12/12
예매할 영화를 선택하세요 : 4

      1      2      3
A      ○      ○      ○
B      ○      ○      ○
C      ○      ○      ○
D      ○      ○      ○
예매할 좌석을 선택하세요(ex) A 1, B 2) : A3
예약이 완료되었습니다.
예약번호는 DCF 입니다.
학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료
메뉴를 선택하세요 : 3

=====예약 확인=====
예약번호를 입력하세요(ex) 예약번호가 AEF라면 입력: AEF): DCF
예매하신 영화 : 청설
예매하신 좌석 : A3

=====영화 예매 좌석=====
      1      2      3

```

먼저, returnManager함수에서 만들어진 MovieManager객체 temp가 생성되면서 MovieManager생성자와 Movie생성자 4개가 출력됩니다. 그리고 MovieManager객체Cha가 returnManager함수에서 반환되는 MovieManager객체 temp를 받아서 생성됩니다. 이때 반환되는 MovieManager객체는 r-value이기 때문에 이동생성자가 생성되고 함수에서 반환된 MovieManager객체는 소멸되서 MovieManager 소멸자가 출력됩니다.

Microsoft Visual Studio 디버그 콘솔

```
=====영화 예매=====
1) 소울 15/15
2) 어벤저스:엔드게임 30/30
3) 너의모든것 25/25
4) 청설 12/12

예매할 영화를 선택하세요 : 4

      1      2      3
A      ○      ○      ○
B      ○      ○      ○
C      ○      ○      ○
D      ○      ○      ○

예매할 좌석을 선택하세요(ex) A 1, B 2) : A3

예약이 완료되었습니다.
예약번호는 DCF 입니다.

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 3

=====예약 확인=====
예약번호를 입력하세요(ex) 예약번호가 AEF라면 입력: AEF): DCF
예매하신 영화 : 청설
예매하신 좌석 : A3

=====영화 예매 좌석=====

      1      2      3
A      ○      ○      ☆
B      ○      ○      ○
C      ○      ○      ○
D      ○      ○      ○
계속하려면 아무 키나 누르십시오 . . .

학번 : 201711430 이름 : 차준혁

=====차준혁 시네마=====
1) 영화 예매 2) 예매 취소 3) 예매 확인 4) 종료

메뉴를 선택하세요 : 4

예매 프로그램을 종료합니다. 안녕히 가세요 ^.^

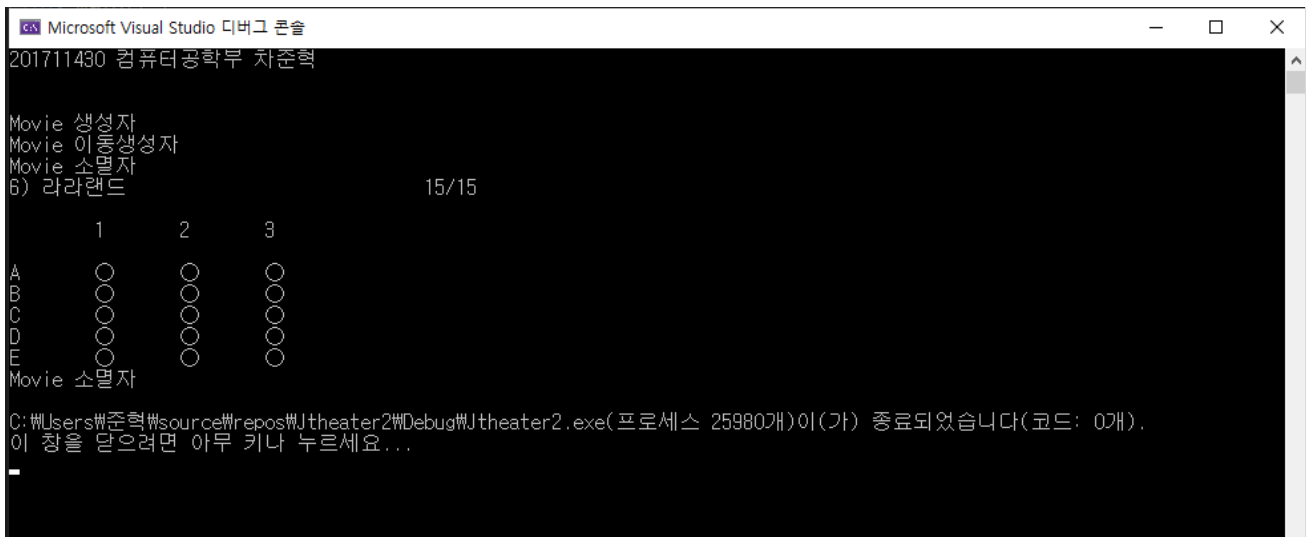
MovieManager 소멸자
Movie 소멸자
Movie 소멸자
Movie 소멸자
Movie 소멸자

C:\Users\차준혁\source\repos\Jtheater2\Debug\Jtheater2.exe(프로세스 13248개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

<Movie객체 이동생성자 test>

```
Movie returnMovie() {  
    Movie temp(5, "라라랜드", 5, 3);  
    return temp;  
}  
cout << "201711430 컴퓨터공학부 차준혁\n" << endl << endl;  
Movie testmovie = returnMovie();  
testmovie.showMovieData();  
testmovie.showSeats();
```

returnMovie함수에서 Movie객체가 생성되고 그 객체를 반환합니다. 그 객체를 받아서 Movie객체 testmovie를 생성합니다. 이 때, returnMovie가 반환하는 Movie객체가 r-value이므로 이동생성자가 생성됩니다.



The screenshot shows the Microsoft Visual Studio Debug Console with the following output:

```
201711430 컴퓨터공학부 차준혁  
  
Movie 생성자  
Movie 이동생성자  
Movie 소멸자  
6) 라라랜드  
15/15  
  
1      2      3  
A      ○      ○      ○  
B      ○      ○      ○  
C      ○      ○      ○  
D      ○      ○      ○  
E      ○      ○      ○  
Movie 소멸자  
  
C:\Users\차준혁\source\repos\theater2\Debug\theater2.exe( 프로세스 25980개)이(가) 종료되었습니다(코드: 0개).  
이 창을 닫으려면 아무 키나 누르세요...
```

4 느낀점 및 토의사항

1차과제에서는 구조체와 3차원 배열을 동적할당해서 과제를 수행했는데 Movie클래스와 MovieManager클래스를 이용해서 객체를 생성해서 다루니 구현하기 훨씬 편했습니다. 저는 스마트 포인터를 사용하지 않았지만 스마트 포인터를 활용하면 동적할당 된 메모리도 알아서 해제할 수 있으니 과제 1보다 더 편하게 구현할 수 있을 것 같습니다. 지금은 txt파일을 읽기만 하지만 추후에 MovieManager객체가 예약한 영화의 좌석 정보도 txt파일에 따로 쓰고 다시 읽고 관리할 수 있는 기능을 만들어도 좋을 것 같습니다.

과제 체크 리스트

1. 과제 제출 목록(보고서, 프로젝트 압축파일)은 모두 제출하였는가?
예
2. 제출한 결과물은 실행가능한가? 예
3. 모든 실행결과물에 학번 이름이 출력되어 있는가? 예
4. Movie의 생성자와 소멸자 테스트 결과를 보고서에 담았는가? 예
5. MovieManager의 생성자와 소멸자 테스트 결과를 보고서에 담았는가? 예
6. 좌석예약 시스템의 모든 기능들이 테스트되어 있는가? 예