객체지향프로그래밍

Assignment 2

학과: 수학과

학번: 20140843

이름: 엄태강

POVIS ID: tkeom0114

Honor Code:나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

1. 개요

이번 과제에서는 C++을 사용하여 영화와 그 정보들을 입력 받은 후, 카테고리 별로 각각의 예산, 수익, 평점의 합, 평균, 최대, 최소를 구해 표와 차트를 출력하는 프로그램을 작성하였다. C++이 객체지향 프로그래밍 언어이므로 이번 과제에서는 영화, 카테고리에 따른 class를 만들어 프로그램을 작성하였다. 추가가능으로 영화를 list에서 삭제할 수 있게 만들었다.

2.class 구성

이번 과제에서는 영화에 대한 정보를 저장하는 class인 movie와 category(genre, source)에 대한 정보를 저장하는 class인 category를 만들었다.

movie

class Movie{

private:

string name;

     string category[2];

     double values[3];

public:

Movie (string,string,string,double,double,double);

string getName (void);

string getCategory (int);

double getValue (int);

};

string name: 해당 영화의 이름을 제작한다.

string category: 해당 영화의 genre, source를 저장한다.

double values: 해당 영화의 production, gross, rating을 저장한다.

Movie (string,string,string,double,double,double): 이름, genre, source, production, gross ,rating을 입력 받으면 그에 해당하는 movie class를 만드는 constructor이다.

string getName (void): 해당 영화의 이름을 return한다.

string getCategory (int): 0을 입력 받으면 해당 영화의 genre를, 1을 입력 받으면 source를 return한다.

double getValue (int): 0을 입력 받으면 해당 영화의 production을, 1을 입력 받으면 gross를, 2를 입력 받으면 rating을 return한다.

category

class Category{

private:

list<Category\*> listOfSecondCategory;

list<Movie\*> listOfMovie;

string category;

int numOfMovies;

double valuesOfCategory[3][4];

public:

Category (Movie,int,const bool);

void addInCategory (Movie,int,const bool);

void deleteInCategory (Movie,int,const bool);

string getCategory (void);

double getValue (int,int);

int getNumOfMovies (void);

list<Category\*> &getList(void);

};

list<Category\*> listOfSecondCategory: genre의 경우 source의 list를, source의 경우 genre의 list를 저장해 세부 category로 접근할 수 있게 한다.

list<Movie\*> listOfMovie: 이 카테고리에 해당하는 영화들의 list이다.

string category: category(genre나 source)의 이름을 저장한다.

int numOfMovies: 이 카테고리에 해당되는 영화의 개수를 저장한다.

double valuesOfCategory[3][4]: 이 카테고리의 각 지표들의 합, 평균, 최대, 최소를 저장한다.

([지표][함수]순이고 지표의 경우 0:production, 1:gross, 2:rating이고, 함수의 경우 0:평균, 1:합, 2:최대, 3:최소이다)

Category (Movie,int,const bool): 추가되는 영화가 새로운 category일 때 category의 종류(genre, source), 본 category인지, 세부 category인지 입력 받아서 category를 만드는 constructor이다.

void addInCategory (Movie,int,const bool):해당 category 내에 movie를 추가하고 각 지표의 합, 평균, 최대, 최소값도 모두 바꾼다.

void deleteInCategory (Movie,int,const bool):해당 category 내에 movie를 제거하고 각 지표의 합, 평균, 최대, 최소값도 모두 바꾼다.

string getCategory (void): 해당 카테고리의 이름을 return한다.

double getValue (int,int): 이 카테고리에 저장된 지표의 통계치 중 원하는 값을 return한다.

(지표, 함수 순으로 입력받고 지표의 경우 0:production, 1:gross, 2:rating이고, 함수의 경우 0:평균, 1:합, 2:최대, 3:최소이다)

int getNumOfMovies (void): 카테고리에 해당되는 영화의 개수를 return한다.

list<Category\*> &getList (void): listOfSecondCategory를 참조하도록 한다.

3.알고리즘

1)시작시

①비어있는 listOfMovie, listOfCategory[2]를 생성한다. listOfCategory 는 각각 genre의 list와 source의 list를 의미한다.

②메뉴를 출력한다.

2)메뉴 입력(int printMenu (void))

①선택할 메뉴를 입력하고 엔터를 누른다

②적절한 입력이면 선택한 메뉴로 넘어가고 부적절한 입력인 경우 경고문을 출력 후 ①을 반복한다.

3)영화 추가(void addMovie (void))

①영화 이름을 입력 받는다.

②이미 리스트에 있는 영화이면 경고문을 출력하고 메뉴로 돌아간다.

③장르, 원작의 종류, 제작비 총액, 수익 총액, 평점을 차례대로입력 받는다. 잘못된 입력의 경우 다시 입력 받도록 한다.

④입력 받은 정보로 movie class를 생성한다.

⑤생성한 movie를 listOfMovie에 삽입한다.

⑥두 개의 category list에 생성한 movie를 삽입한다. (항목5)참고)

4)영화 삭제(void deleteMovie (void))

①movie list가 비어있는 경우 바로 경고문을 출력하고 메뉴로 돌아간다.

②영화 이름을 입력 받는다.

③리스트에 없는 영화이면 경고문을 출력하고 메뉴로 돌아간다.

④listOfMovie내에서 같은 이름을 가진 영화를 찾는다.

⑤새로운 movie class를 만들어 그 영화의 이름, 장르, 원작의 종류를 복사한다. 이때 실수 값들은 모두 0으로 둔다.

⑥listOfMovie에서 찾은 영화를 삭제한다.

⑦두 개의 category list에서 해당 movie를 찾아 제거한다. (항목 6)참조)

5)category list에 영화 삽입 (void addMovieInList (list<Category\*> \*listOfCategory, Movie movie, int selectCategory,const bool doingRecursive))

①listOfCategory를 순회하면서 movie가 해당하는 category를 찾는다.

②category를 발견하면 그 카테고리에서 addInCategory를 호출해 movie를 삽입한다.

③category를 발견하지 못하면 대소문자를 무시한 사전 순으로 category의 위치를 찾고 해당 위치에 constructor로 만든 category를 listOfCategory에 삽입한다.

6)category list에서 영화 제거 (void deleteMovieInList (list<Category\*> \*listOfCategory, Movie movie, int selectCategory,const bool doingRecursive))

①listOfCategory를 순회하면서 movie가 해당하는 category를 찾는다.

②category를 발견하면 그 카테고리에서 deleteInCategory를 호출해 movie를 제거한다.

7)영화 목록 출력(void printList(void))

①”Name Genre Source Production Gross Rating”을 출력한다.

②listOfMovie를 순회하면서 각 영화의 정보들을 한 줄에 출력한다.

8)pivot table 출력(void pivotTable(void), void printList (list<Category\*> \*listOfCategory, int figure, int function, const bool doingRecursive))

①출력할 카테고리, 출력할 항목, 원하는 통계(평균, 합, 최대, 최소)를 입력받는다.

②1개의 카테고리만 출력하는 경우 그 카테고리의 listOfCategory, 아닌경우는 첫 번째 카테고리의 listOfCategory를 pivotTable의 인자로 넣어 호출한다.

③한 카테고리만 입력되면 printList의 doingRecursive에 0을, 2개의 카테고리를 모두 이용하는 경우 doingRecursive에 1을 인자로 넣어 호출한다.

④doingRecursive가 0이면 해당 listOfCategory의 선택한 항목의 통계를 category를 순회하면서 출력한다.(사전 순으로 배열되어 있어 출력되 사전 순이다.)

⑤ doingRecursive가 1이면 listOfCategory의 category들을 순회하면서 각 category의 listOfSecondCategory를 printList의 인자로 사용해 호출한다.

9)pivot chart 출력(void pivotChart(void))

①출력할 카테고리, 출력할 항목, 원하는 통계(평균, 합, 최대, 최소)를 입력받는다.

②1개의 카테고리만 출력하는 경우 doingRecursive를 0, 2개의 카테고리에 대해 출력하는 경우 doingRecursive를 1로 둔다.

③ listOfCartegory의 각 list의 element의 개수를 numOfGenre, numOfSource에 저장한다.

④ string의 배열을 2개 생성해(string indexOfGenre,indexOfSource) 각각 genre와 source들의 이름을 저장한다. listOfCategory내에 genre와 source는 정렬되어 있으므로 index내에도 정렬된 상태로 저장된다)

⑤선택된 항목과 통계치를 모든 category와 비교해 최댓값과 최솟값을 찾는다.

⑥listOfCategory의 각 항목을 탐색하면서 index string을 이용해 index를 출력하고 위에서 구한 최댓값과 최솟값을 이용해 =를 출력한다.

⑦chart를 모두 작성하면 아래에 index와 index에 해당하는 category를 출력한다.

10)프로그램 종료

①동적으로 할당된 movie와 category들을 모두 제거하며 종료한다.

4.C로 구현한 것과의 차이

C언어에는 class라는 개념이 없다. 하지만 이와 비슷하게 관련된 데이터들을 묶을 수 있는데 이것이 바로 struct이다. 따라서 각각의 영화, 장르, 출처에 대해 movie, genre, source라는 struct를 만들었다. 그리고 C에서는 새로운 struct를 정의할 때 scope를 벗어나서도 값이 변경되기를 원하면 malloc을 사용해 직접 메모리 공간을 할당했지만 C++에서는 new를 사용하여 class의 동적 할당을 할 수 있었다.

C++에서는 list를 STL에서 가져와서 쓸 수 있지만 C는 그렇지 않기 때문에 movie, genre, source에 대해 list struct를 구현했다.

또한, C에서는 struct의 원소에 직접 접근해서 함수를 변경했지만 C++에서는 class 내의 data를 private로 두어 보호하고 class의 method를 통해 수정하게 만들었다.

5.실행방법

make 후 out파일을 실행하면 메뉴가 아래와 같이 뜬다.

1.add movie

2.print list

3.pivot table

4.pivot chart

5.remove movie

6.exit

“selection: “에서 1을 입력하면 영화 추가, 2를 입력하면 영화의 리스트 출력, 3을 입력하면 pivot table출력, 4를 입력하면 pivot chart를 출력, 5는 영화 제거, 마지막으로 6은 프로그램을 종료한다.

영화를 추가할 때는 제목, 장르, 원작의 출처, 제작비, 수익, 평점을 입력하면 된다.

2를 선택하면 바로 영화들의 모든 정보를 포함한 리스트를 출력한다.

3을 선택할 경우 아래와 같이 분류하기를 원하는 카테고리를 선택하게 한다.

1.genre

2.source

3.genre and source

4.source and genre

1,2의 경우 각각 genre와 source에 따라 분류하고, 3,4는 두 카테고리를 모두 사용해서 분류하되 3은 genre에 대해 먼저 분류하고, 4는 source에 대해 먼저 분류한다.

그 다음 아래와 같이 어떤 항목을 나타낼 것인지 선택하게 한다.

1.Production

2.Gross

3.Rating

1은 제작비, 2는 수익, 3은 평점에 대해 출력하게 한다.

그 다음 아래와 같이 어떤 통계치를 출력할 것인지 선택하게 한다.

1.Average

2.Sum

3.Max

4.Min

1은 category내의 평균, 2는 category내 모든 영화의 총합, 3은 최대, 4는 최솟값을 출력하게 한다.

모든 선택이 다 끝나면 선택에 따라 pivot table을 출력한다.

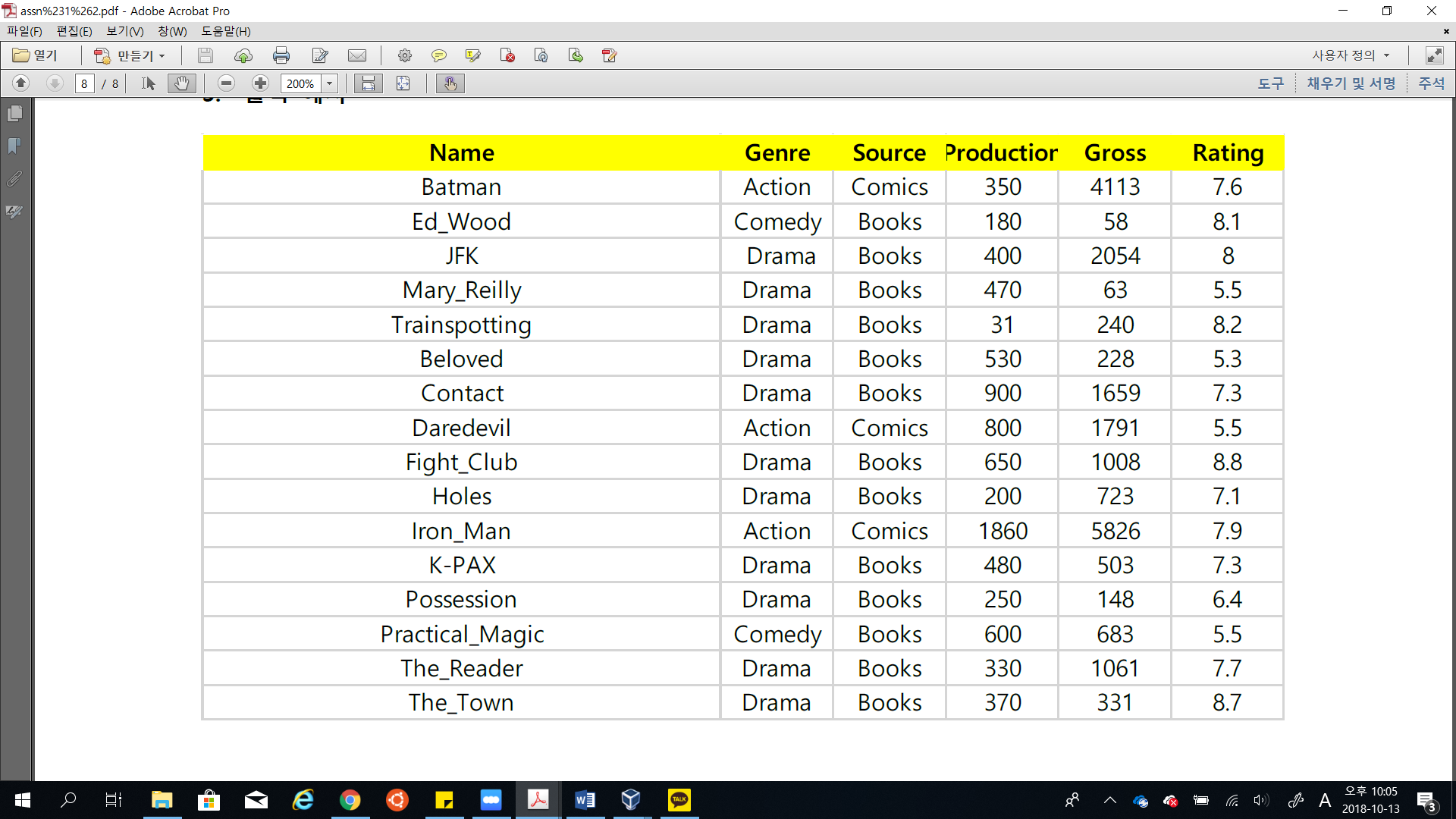
메뉴에서 4를 선택하면 3에서와 같이 카테고리, 항목, 통계치를 선택한 후 선택된 pivot chart를 출력한다.

메뉴에서 5를 선택하면 제거할 영화의 이름을 입력해서 영화를 제거하면 된다.

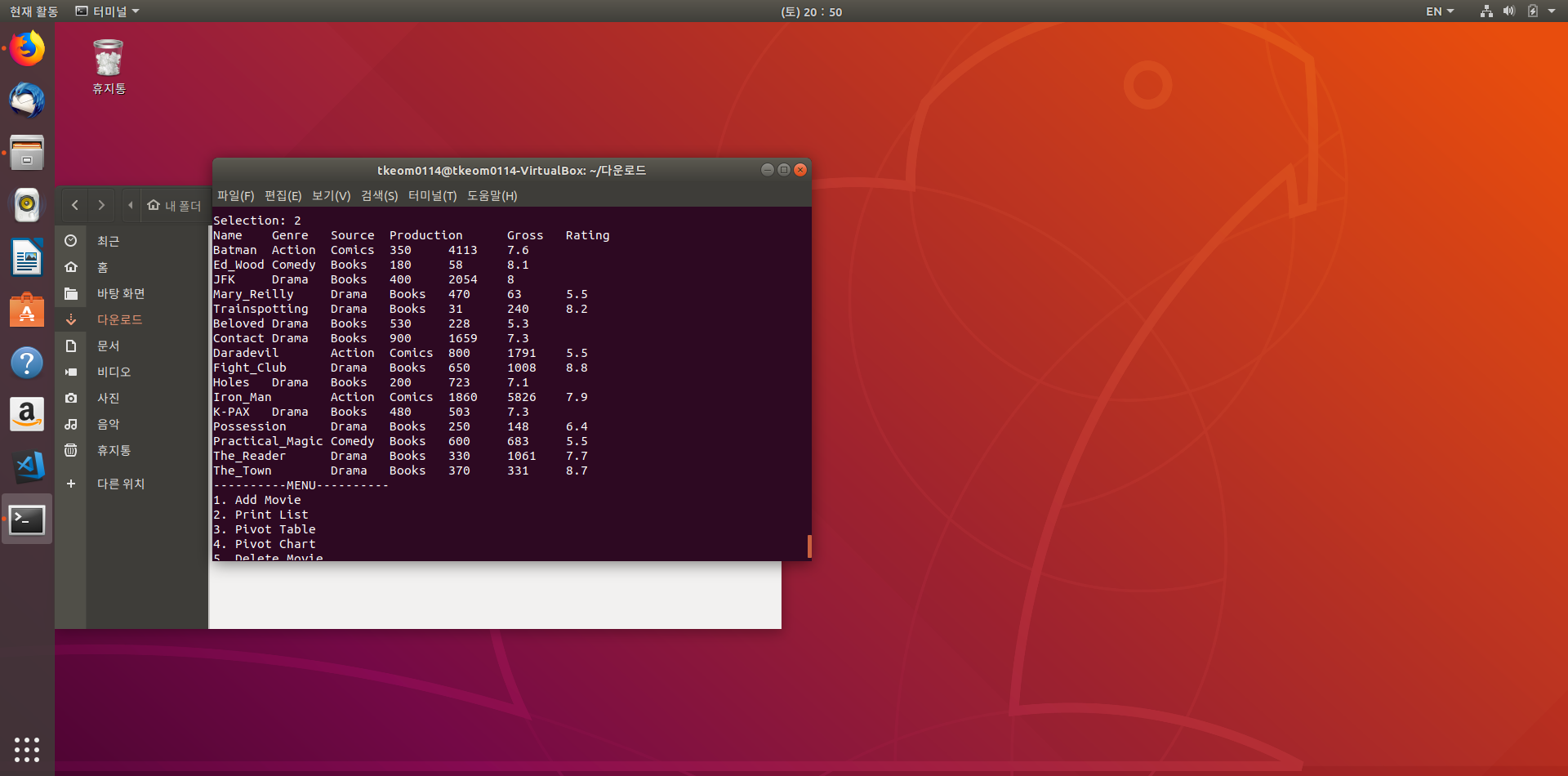
프로그램 종료 시 6을 선택하면 된다.

6.실행예시

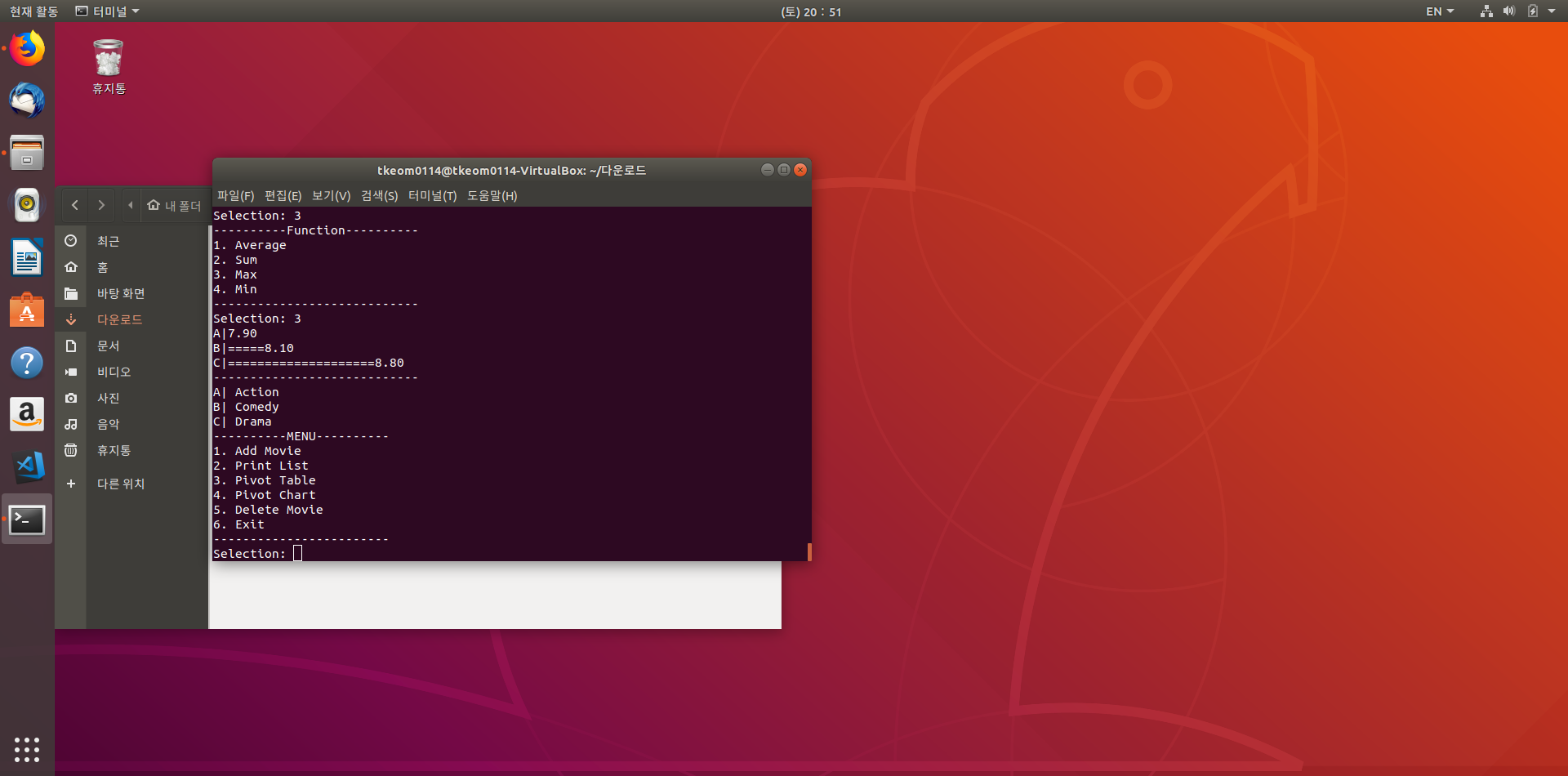
다음과 같이 입력한다.



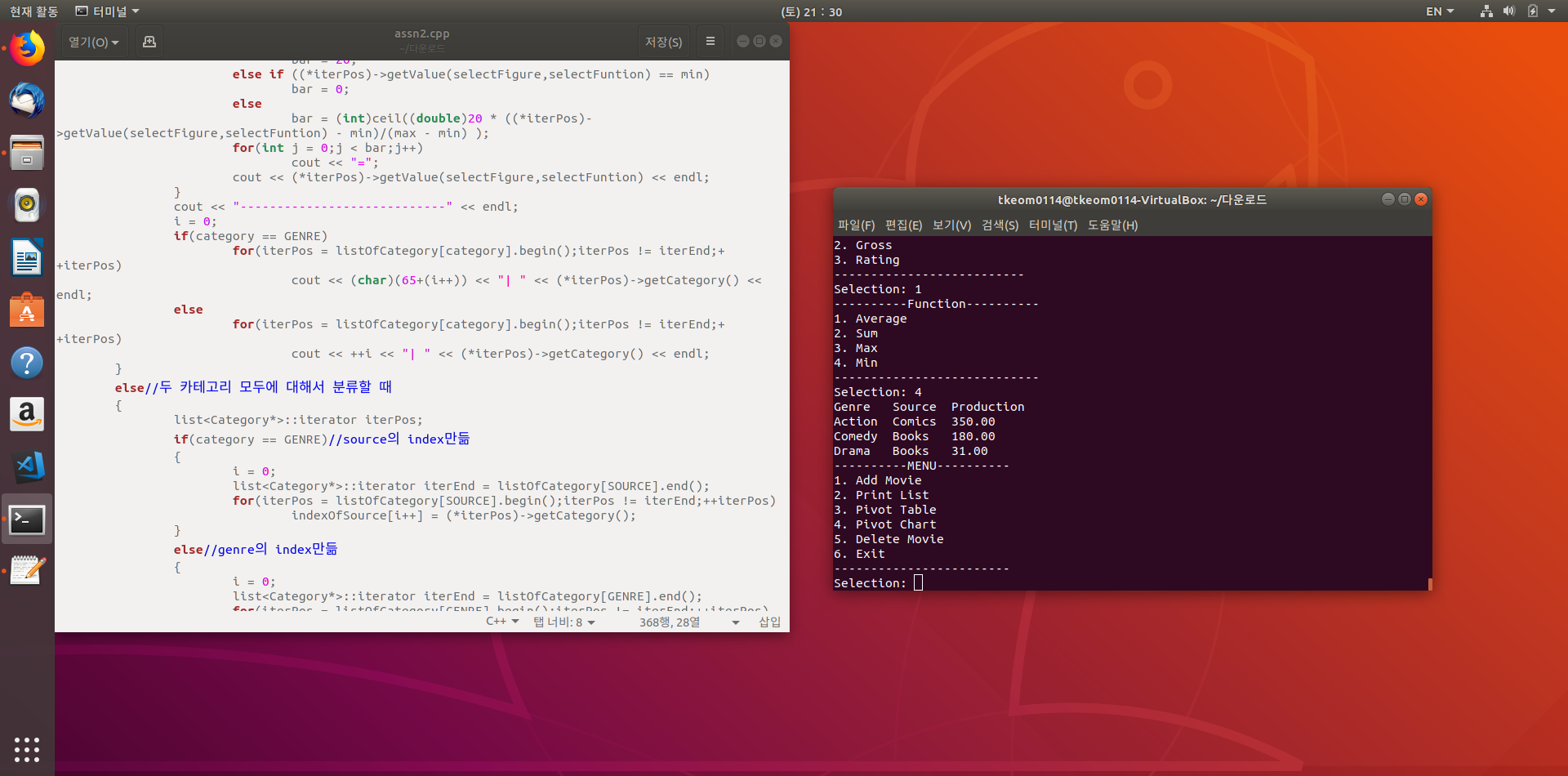
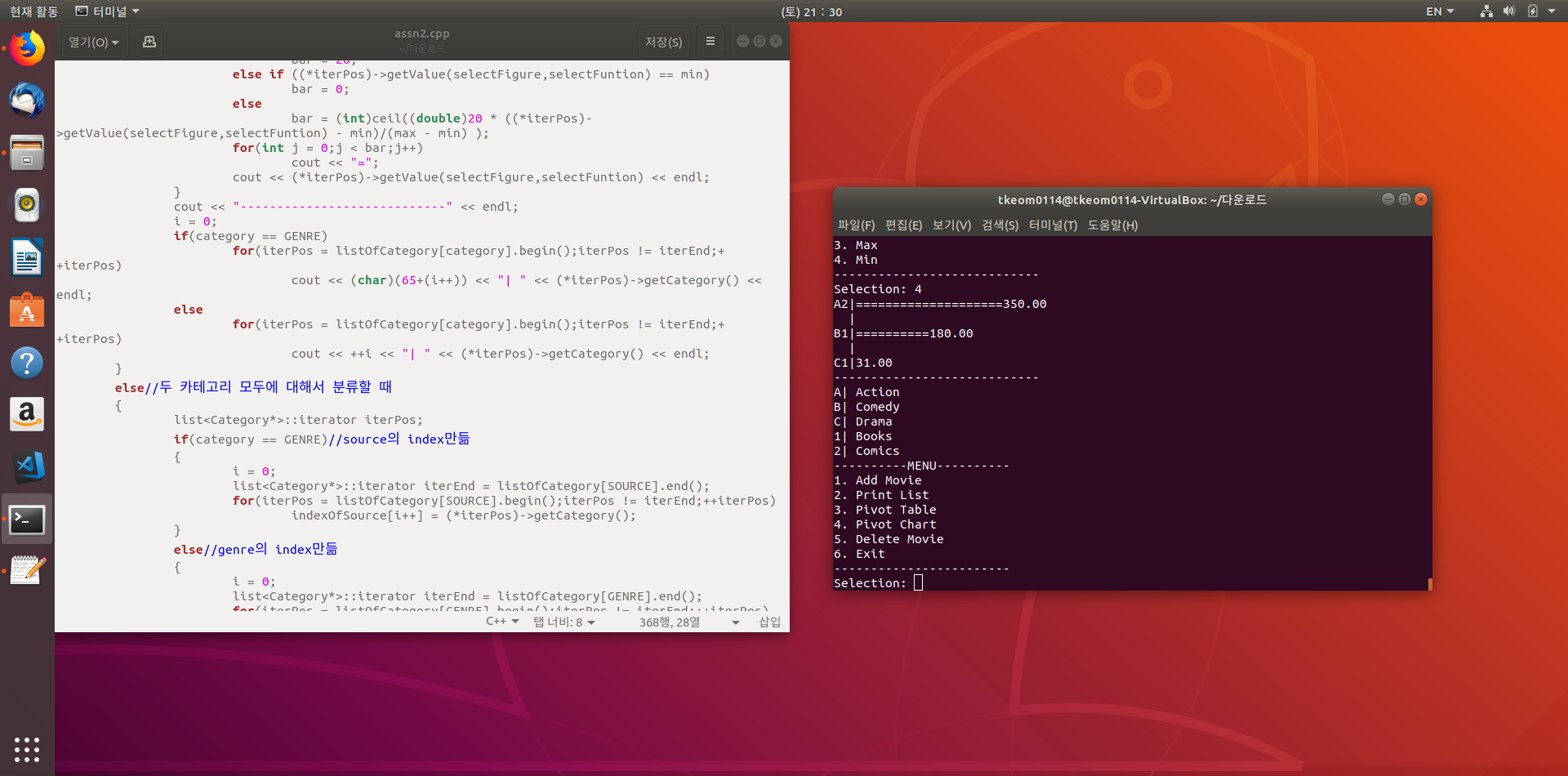
list를 출력한 결과는 다음과 같다.



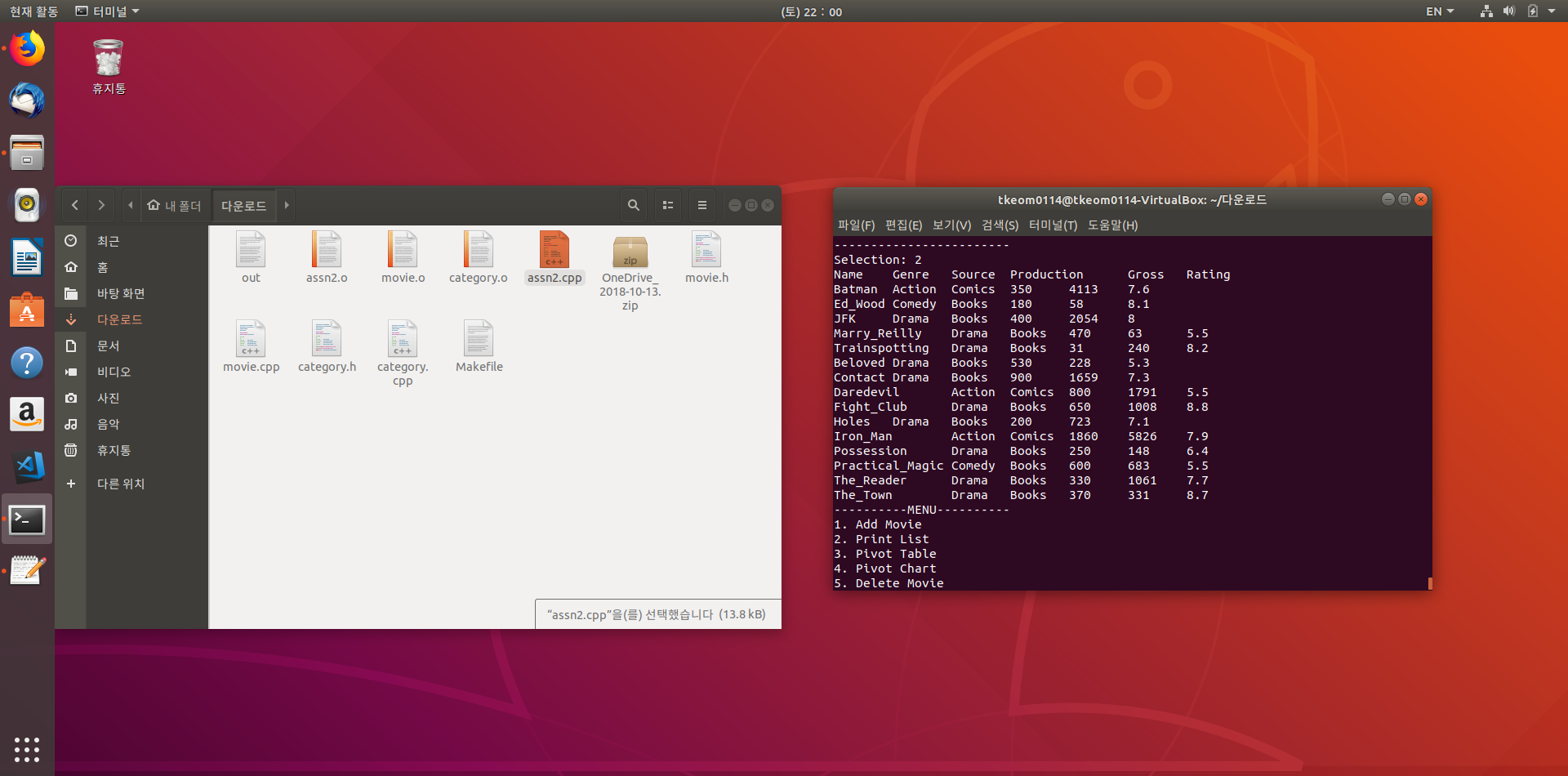
Genre별 Max Rating 차트를 출력한 결과는 다음과 같다.



Genre와 Source조합별 min production 차트와 테이블을 출력한 결과는 다음과 같다.



위 리스트에서 K\_PAX를 삭제한 결과는 다음과 같다.



7.토론

1)잘못된 입력의 처리

    do{

        cout << "Selection: ";

        cin >> select\_menu;

        if(cin.fail())

        {

            cout << "Try Again!" << endl;

            cin.clear();

            cin.ignore(1000,'\n');

            continue;

        }

        if (select\_menu < 1 || select\_menu > 6)

            cout << "Try Again!" << endl;

    } while (select\_menu < 1 || select\_menu > 6);

정수를 입력해야 할 경우 범위를 벗어난 정수이면 범위만 if문으로 체크해 다시 입력 받으면 된다. 하지만 아예 문자를 입력해 버린 경우 범위를 체크하지 못하고 정수가 들어가야 할 메모리 공간에 이상한 값이 들어가게 된다. 이 경우 cin.fail을 사용해 에러를 감지해서 플래그 값을 변경하고, cin.clear를 이용해 플래그를 정상화한다. 그 다음 cin.ignore를 사용해 버퍼에 잘못 입력된 값을 무시할 수 있게 한다. 이 내용은 [1]에서 참고했다.

2)포인터의 list를 사용한 이유

리스트에서 영화를 제거할 경우, delete를 사용해 메모리를 해제해야 해서 그 객체의 주소를 알아야 한다. 이 과제에서는 STL을 사용했기 때문에 iterator를 이용해서 해당 객체에 접근하였고 이 경우, iterator에서는 public으로 설정한 method들밖에 접근할 수 없어 객체의 주소를 얻을 수 없었다. 따라서 객체들로 직접 list를 만들지 않고 객체들의 주소들로 list를 만들어 delete(\*iterator)와 같은 방식으로 해당 객체를 메모리 해제하였다. 이 내용은 [2]에서 참고했다

8.결론

이번 과제에서는 C++을 사용하여 영화와 그 정보들을 입력 받은 후, 카테고리 별로 각각의 예산, 수익, 평점의 합, 평균, 최대, 최소를 구해 표와 차트를 출력하는 프로그램을 작성하였고 추가기능으로 영화를 삭제할 수 있게 구현했다. C++을 사용해서 프로그램을 처음 작성해 보았는데

객체지향 프로그래밍이라는 패러다임에 익숙하지 않아 C에 비해 편한 부분도 많이 있었지만 어색하고 불편한 부분도 없지않아 있었다. 그리고 C에서와는 달리 주소에 직접 접근하는 것을 최소화 하기 위해 참조를 쓰는 것이 일반적이고, 처음 구현할 때 그렇게 시도했다. 하지만 영화 제거 기능을 구현하면서 포인터 대신 참조를 사용하는데 실패하였고 결국 포인터를 이용해 프로그램을 구현한 것에 대해 아쉬움이 남는다. 공부하면서 이에 대한 개선점을 찾을 수 있었으면 좋겠다.

9.Reference

[1] <https://stackoverflow.com/questions/5131647/why-would-we-call-cin-clear-and-cin-ignore-after-reading-input>

[2] https://stackoverflow.com/questions/42959192/list-of-subclass-objects-reinterpreted-as-a-list-of-base-class-objects-c11