**전체화면 및 우측 하단 인쇄 모양 보기로 봐주세요!**



객체지향프로그래밍

Report #7

|  |  |
| --- | --- |
| 학번 | 20174627 |
| 이름 | 김혜진 |

문제 명세

**● 주제 선택**

* Library System
  + 도서 관리(조회, 삽입, 삭제 기능 등)
  + 도서 대여, 회원 등록(회원 구분: 학생, 대학원생, 교수 등
* Movie World
  + 영화 DVD 관리(조회, 삽입, 삭제 기능)
  + 대여 기능, 회원 등록(회원 구분: 일반, 골드 회원 등)
* Product Inventory System (여러분이 선호하는 품 목 선택)
  + 물품 관리 (조회, 삽입, 삭제 기능 등)
  + 구매 기능, 회원 등록(회원 구분)

**● C++ 요구 기능**

* 상속
* 연산자 중복
* 동적 바인딩(가상 함수)
* 예외 처리
* 포괄적 함수
* 그 외
* **지침 사항**
* 보고서는 문제 명세, 분석, 설계, 구현, 테스트/검증, 의견 순으로 작성
* 분석에는 클래스 계층구조객체 관계 다이어그램 포함
* 물품(도서, 영화, 품목)과 회원의 클래스 설계시에 반드시 클래스 계층 구조를 고려해야 한다.
* 설계에는 클래스 설계, 함수 알고리즘 기술 포함

물폼 목록은 파일상에 관리 (파일 읽기/쓰기 포함) (여기까지15점)

* 부가 기능은 사용자 편의성 제공을 위한 제반 기능을 포함한다(GUI 등 프로그램 기능 이용 편의성 제공, 스마트폰 어플 등을포함, GUI 이외 코드는 반드시 C++로 작성해야 함) (5점)

C++ 기능표

|  |  |  |
| --- | --- | --- |
| **기능** | **내용** | **사용 분야** |
| **C++ 기본** | 포인터 | 동적 할당 시 구조체포인터 및 클래스포인터 사용 |
| 배열 및 구조체 | 영화 정보를 입력하는 Movie 구조체 사용 |
| 레퍼런스 | MENU 함수에서 좌표를 지정하는 x,y에 사용 |
| **동적메모리 할당** | New, delete | Movie , Reservation , VIP\_Reservation를  동적 할당, 해제할 때 사용 |
| **예외처리** | Try, catch문 | 예외처리가 필요한 구간에 Try, catch문 사용으로 예외처리 |
| **포괄적 함수** | template | 호출 시 매개변수의 타입별로 실제 함수를 작성 |
| **개선된 함수** | 디폴트인자 | check 함수에서 사용 |
| 인라인 함수 및  매크로 함수 | PrintMovie에서 사용 |
| 함수오버로딩 | check 함수에서 사용 |
| **문자열** | C 및 C++  문자열의 사용 | Movie의 변수에 사용 |
| 문자/문자열 입출력 | cout , cin 빈번하게 사용 |
| **클래스 기초** | 클래스 설계 | Reservation, VIP\_Reservation에 클래스 사용 |
| 분할 컴파일 | 각 클래스와는 헤더와 cpp를 따로 구성 |
| **생성자 및 소멸자** | 디폴트 생성자 | 클래스들이 멤버변수를 초기화 해주는 용도로 사용 |
| 인자 있는 생성자 | 클래스들은 복수의 인자있는 생성자를 가지고 있음 |
| 소멸자 | 클래스들이 소멸자가 호출되면 화면을 지우고 안내메세지를 출력해주는 용도로 사용 |
| const 객체와  const 멤버함수 | 전달 인자에 const 객제 사용, 객제의 보호가 보장 |
| **여러 가지 객체의**  **생성 방법** | 동적객체, 객체배열,  객체에 대한  포인터 배열 등 | Movie \*M = new Movie[max]  Reservation \*R = new Reservation[max]; 등 동적 객체 사용 |
| **상속과 다형성** | 상속 | Reservation 클래스를 상속받은 VIP\_Reservation 클래스에 사용 |
| 동적바인딩 | Reservation 클래스와 상속받은 VIP\_Reservation 클래스의  void place\_situation(int x, int y)를 동적바인딩(가상함수) |
| **그 외** | gotoxy(int x, int y) | 좌표를 옮겨주는 함수 |
| system("cls") | 전체화면을 초기화 해준다 |

분석

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **Struct Movie** | | **영화 정보를 저장하는 구조체** | | string stdio; // 영화사 이름  string now\_showing; // 현재 상영 중인 영화  string coming\_soon; // 개봉 예정중인 영화 | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **Reservation** | | **일반 예매를 하는 클래스** | | int remain(); // 남아있는 좌석의 개수를 리턴해주는 함수  void ticket(); // 선택한 좌석을 예매하는 함수  void place\_situation(); // 현재 좌석 상황을 표시하는 함수  int USE(); // 해당 좌석이 사용 중인지 확인하는 함수 | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **VIP\_ Reservation** | | **VIP 회원으로 영화를 예매하는 클래스 - Reservation 클래스 상속** | | void discount(); // 할인쿠폰을 적용하는 함수  void place\_situation(); // 좋은 자리와 현재 좌석의 상황을 표시하는 함수 | |

|  |
| --- |
|  |

설계(설명은 주석)

|  |  |
| --- | --- |
| **main.cpp** | **struct Movie (구조체)** |
| #include "R.h"  typedef struct Movie{  string stdio; // 영화사 이름  string now\_showing; // 현재 상영중인 영화  string coming\_soon; // 개봉 예정중인 영화  }Movie;  int check(Reservation\* R, int m , int L, int C); // 좌석 범위 체크  int check(Reservation\* R, int m , char L, int C); // 좌석 범위 체크  void InputMovie(Movie \*M, int n);  void inline PrintMovie(Movie \*M, int n , Reservation\* R);  void MENU(int &x, int &y);  int main()  {  int max = MAX; // 이미 할당된 배열의 최대 크기를 세는 변수  int num =0;  int N;  /////////////////////////////////////////////////////////////////////////  cout << " 영화정보를 입력하세요";  cout << endl;  Movie \*M = new Movie[max]; // 영화사 4개까지 입력  while(1)  {  if(num == max)  {  max +=1;  Movie \*temp = new Movie[max]; // 할당된 크기만큼 새로운 구조체포인터 temp 동적할당  memcpy (temp,M,(num)\*sizeof(Movie)); // 문자열 복사기능  //delete M; // 동적으로 할당한 M 할당 해제  M = temp;  }  for(int i = num; i<max ; i++)  {  InputMovie(M,i);  num++;  }    cout << "영화정보를 더 입력하시겠습니까? (1.Yes, 2.No)";  cin >> N;  cout << "\n";    if(N==2) // NO 입력시 무한루프 종료  break;  } // while 종료  cout << "영화정보를 모두 입력하셨습니다. "<< endl;  cout << "영화 예매 프로그램을 시작하겠습니다." << endl;  Sleep(700);  system("cls");  ///////////////////////////////////////////////////////////////////////  int X=15,Y=4+num ,x,y;  int menu;  int m;  Reservation\* R = new Reservation[max]; // 영화 갯수만큼 R 동적할당  VIP\_Reservation \*VIP\_R = new VIP\_Reservation[max]; // 영화 갯수마다 VIP\_R 동적할당  while(1) // 무한루프  {  PrintMovie(M,num,R); // 영화 정보 출력  x=X,y=Y; // 좌표값 변수  MENU(x,y); // 메뉴 출력  cin >> menu;    if(menu == 1) // 일반 영화 예매  {  char L; // 행  int C; // 열  y+=2;  gotoxy(x,y);  cout << "영화관의 번호를 입력해주세요 : ";  cin >> m;  m--;  y+=2; x+=6;  R[m].place\_situation(x,y); // m관의 영화 자리 표시  cout << "좌석을 입력해주세요(ex> A,0) : ";  cin >> L >> C;  if(check(R,m,L,C) == 99) // 입력된 좌석이 범위에 맞지않으면  {  while(check(R,m,L,C) == 99) // 맞지 않으면 계속 반복  {  gotoxy(x,y+7);  cout << " "; // 표시된 좌석을 지워주는 역활  gotoxy(x,y+7);  cout << "좌석을 다시 입력해주세요 : "; // 좌석 다시 입력받기  cin >> L >> C;  check(R,m,L,C); // 범위에 맞는지 계속 체크  }  }  // 범위에 맞으면  R[m].ticket(L,C); // 해당 좌석 티켓팅  R[m].place\_situation(x,y); // 바뀐 좌석 출력  gotoxy(x,y+9);  cout << "예매가 완료되었습니다." ;  gotoxy(x-1,y+10);    if(M[m].coming\_soon != "X") // 개봉영화가 있다면  { // 해당 stdio 의 개봉영화 홍보  cout << M[m].stdio <<"의 '"<<M[m].coming\_soon <<"'이 개봉예정입니다.";  gotoxy(x,y+11);  cout <<"많은 관심 부탁드립니다.";}  VIP\_R[m].ticket(L,C); // VIP 클래스 같은자리에 티켓팅 (두개의 클래스의 좌석이 티켓팅)  }  else if(menu == 2) // VIP 영화 예매  {  char L;  int C;  y+=2;  gotoxy(x,y);  cout << "영화관의 번호를 입력해주세요 : ";  cin >> m;  m--;  y+=2; x+=6;  VIP\_R[m].place\_situation(x,y); // VIP 좌석 표시 (+기능) 좋은자리 표시  cout << "좌석을 입력해주세요(ex> A,0) : ";  cin >> L >> C;    if(check(R,m,L,C) == 99) // 입력된 좌석이 범위에 맞지않으면  {  do  {  gotoxy(x,y+7);  cout << " ";  gotoxy(x,y+7);  cout << "좌석을 다시 입력해주세요 : ";  cin >> L >> C;  check(R,m,L,C);  }while(check(R,m,L,C) == 99);  }  // 좌석이 범위에 맞으면  VIP\_R[m].ticket(L,C); // 해당 자리 티켓팅  VIP\_R[m].place\_situation(x,y); // 바뀐 좌석 출력  gotoxy(x-4,y+9);  VIP\_R[m].discount(); // 할인쿠폰 적용 (VIP클래스 전용)  y++;  gotoxy(x,y+10);  cout << "예매가 완료되었습니다." ;  gotoxy(x-1,y+11);  if(M[m].coming\_soon != "X") // 개봉영화가 있다면  { //해당 studio의 개봉영화 홍보  cout << M[m].stdio <<"의 '"<<M[m].coming\_soon <<"'이 개봉예정입니다.";  gotoxy(x,y+12);  cout <<"많은 관심 부탁드립니다.";}  R[m].ticket(L,C); // 일반자리도 티켓표시    }  else // 종료  {  gotoxy(X+4,Y+4);  cout << "프로그램이 종료됩니다.";  break;  }  Sleep(3000); // 3초 정지 후  system("cls"); // 화면 초기화  }  Sleep(1000); // 무한루프를 벗어나면 1초 멈춤  delete[] R; // 메모리 해제  delete[] VIP\_R; // 메모리 해제  //소멸자 호출  }  void MENU(int &x, int &y) // 좌표가 다같이 바뀌어야하므로 레퍼런스로 매개변수  {  gotoxy(x,y);  cout << "1. 영화 예매하기" ;  y++;  gotoxy(x,y);  cout << "2. VIP회원으로 영화 예매하기" ;  y++;  gotoxy(x,y);  cout << "3. 종료"<< endl;  gotoxy(x+30,y);  cout << "\n" << "\n" << " " << "☞ ";  }  void gotoxy(int x, int y) // 커서의 위치를 이동시키는 함수  {  COORD Pos = {x-1, y-1};  SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE),Pos);  }  void InputMovie(Movie \*M, int n)  {  cin.clear(); // 버퍼 비워주기  fflush(stdin); // 버퍼 비워주기  cout << "▶ stdio : ";  getline(cin,M[n].stdio);  cout << "▶ now\_showing : ";  getline(cin,M[n].now\_showing);  cout << "▶ coming\_soon : ";  getline(cin,M[n].coming\_soon);  cout <<"\n";  }  void inline PrintMovie(Movie \*M, int n, Reservation\* R)  {  for(int i=0;i<n;i++) // 첫번째 고객부터 입력한 고객번호까지 출력  {  gotoxy(15,3+i);  cout << (i+1) <<"관 " ;  cout.setf(ios\_base::left); // 왼쪽정렬  cout.width(10); // 너비 10  cout << M[i].stdio ;  cout.width(12);  cout << M[i].now\_showing ;  gotoxy(47,3+i);  cout << R[i].remain();  }  }  int check(Reservation\* R, int m , int L, int C=0)  {  return 99;  }  int check(Reservation\* R, int m , char L, int C) // 오류 (문자만 입력한 경우)  {  if(L < 65 || L >68) // A~D 범위에 벗어난경우  return 99;  if(C<1 || C>9) // 1~9 범위에 벗어난경우  return 99;  int use = R[m].USE(L, C); // 입력된 자리 사용여부 확인  if(use ==1) // 입력된 자리가 사용한 자리라면  return 99;  } // 99는 오류로 main에서 판정 | |

|  |
| --- |
| **Reservation.cpp** |
| #include "R.h"  void Reservation::ticket(const char l,int r) // 해당 좌석을 예매하는 함수  {  r--; // 배열은 0부터 시작이므로 -1 해준다.  if(l == 'A') // A는 0부터 시작  r;  else if(l == 'B') // B줄은 10부터 시작  r+=10;  else if(l == 'C') // C줄은 20부터 시작  r+=20;  else if( l =='D') // D줄은 30부터 시작  r+=30;  else // A~D 범위가 아닐경우 오류 메세지  cout << "오류" ; // 출력  place[r] = 1; // 해당자리 사용(1)으로 표시  able--; // 현재 좌석 갯수 -1  }  int Reservation::remain() // 현재 남아있는 좌석 갯수를 리턴하는 함수  {  return able; // 남은 좌석갯수 리턴  }  void Reservation::place\_situation(int x, int y) // 현재 좌석 상황 표시하는 함수  {  char alpa = 65; // A~D 문자를 나타낼 변수  int j=0;  gotoxy(x+5,y); // 커서 이동  cout << "S C R E E N";  y++;  gotoxy(x,y);  for(int i=0;i<40;i++)  {  if(i%10 == 0)  {  if(j==0)  cout << " 1 2 3 4 5 6 7 8 9 10";  y++;  gotoxy(x,y);  cout << alpa;  alpa++;  j++;  }  if(place[i]==0) // 사용 안함(0) 자리는 □ 으로 표시  cout << "□";  else if (place[i] ==1) // 사용함(1) 자리는 ■ 으로 표시  cout << "■";  }  gotoxy(x,y+2);  }  int Reservation::USE(char l, int r) // 해당자리가 사용중인지 리턴해주는 함수  {  r--;  if(l == 'A')  r;  else if(l == 'B')  r+=10;  else if(l == 'C')  r+=20;  else if( l =='D')  r+=30;  return place[r]; // 해당 자리 상태 출력 (사용함 1, 사용안함 0)  } |

|  |
| --- |
| **VIP\_Reservation.cpp** |
| #include "R.h"  void VIP\_Reservation::discount() // 랜덤으로 쿠폰 할인 적용  {  cout <<"(VIP)"<< coupon <<"% 할인된 가격으로 적용됩니다.";  }  void VIP\_Reservation::place\_situation(int x, int y) // 현재 좌석의 상황을 표시하는 함수 (재정의)  {  char alpa = 65;  int j=0;  gotoxy(x+5,y);  cout << "S C R E E N";  y++;  gotoxy(x,y);  for(int i=0;i<40;i++)  {  if(i%10 == 0)  {  if(j==0)  cout << " 1 2 3 4 5 6 7 8 9 10";  y++;  gotoxy(x,y);  cout << alpa;  alpa++;  j++;  }  if(place[i]==0) // 사용 안함(0)은 □ 으로 표시  cout << "□";  else if (place[i] ==1) // 사용중(1) 은 ■ 으로 표시  cout << "■";  else if (place[i] == 9) // 좋은자리는 ☆ 으로 표시  cout << "☆";  }  gotoxy(x,y+2);  } |

|  |  |  |
| --- | --- | --- |
| **R.h** | **Reservation 클래스** | **VIP\_Reservation 클래스** |
| #include <iostream>  #include <string>  #include <Windows.h>  using namespace std;  #define MAX 2  ////////////////////////////////////////////////////  void gotoxy(int x, int y);  //////////////////////////////////////////////////  class Reservation  {  protected:  int place[40]; // 좌석 표시  int able ; // 남은 좌석의 갯수  public:  Reservation() // 디폴트 생성자  {  able=40; // 남은 좌석을 0 으로 초기화  for(int i=0 ; i<40 ; i++)  place[i] =0; // 좌석을 사용안함(0)으로 초기화  }  Reservation(int able\_num)  {  able = able\_num ;  for(int i=0 ; i<able ; i++)  place[i] =0; // 좌석을 사용안함(0)으로 초기화  }  ~Reservation() // 소멸자  {  system("cls");  gotoxy(20,5);  cout << "이용해 주셔서 감사합니다.";  gotoxy(20,6);  }  int remain() ; // 남아있는 좌석 출력  void ticket(const char l ,int r); // 좌석을 예매하는 함수  void place\_situation(int X, int Y); // 현재 좌석 상황 표시  int USE(char l, int r); // 해당 좌석이 사용중인지 확인  };  class VIP\_Reservation : public Reservation  {  protected:  int coupon;  public:  VIP\_Reservation()  {  coupon = rand()%3+1;  coupon = coupon\*10; // 10 20 30% 랜덤으로 할인  for(int i=0 ; i<40 ; i++)  place[i] =0; // 좌석을 사용안함(0)으로 초기화  for(int j=0 ; j<4 ; j++)  {  place[23+j] = 9; // 좋은자리(9) 표시  place[33+j] = 9;  }  }  void discount();  void place\_situation(int X, int Y); // (+기능)좋은자리 표시 }; | | |

알고리즘 설명 및 구현

|  |
| --- |
| 1. 프로그램 실행 2. 상영할 영화 정보 입력 받음      1. 1번을 입력하면 계속해서 영화 정보를 입력 받고, 2번을 입력하면 위와 같이 프로그램을 시작하겠다는 문구를 띄운 후 예매 프로그램으로 넘어감      1. 예매 화면에서 어떤 회원으로 예매할 것인지 번호를 선택함      1. 회원 선택 후, 예매할 영화관의 영화관 번호를 입력 받음      1. 번호를 입력 받으면 현재 그 영화의 좌석 상황을 출력함      1. 선택할 좌석을 입력 받음 2. 선택한 좌석이 문제가 없다면 해당 좌석을 예매함      1. 만약 할당되어 있는 자리 외의 다른 자리를 입력했다면 좌석을 다시 입력하라는 문구 출력      1. 예매가 완료되면 좌석 색이 변하고, 예매 완료 메시지가 출력됨.      1. 예매가 끝나면 예매된 영화 만큼의 숫자가 줄어들고 다시 초기화면으로 돌아감      1. 만약, VIP 회원으로 예매하고 싶다면 VIP 회원으로 예매하기를 선택함   이는 일반 예매와 마찬가지로 예매할 영화관 번호를 입력 받음   1. 단, 일반 예매와는 다르게 VIP 예매는 프리미엄 좌석을 보유하고 있기 때문에 이를 표시 해줌      1. 일반 예매와 마찬가지로 좌석을 입력 받음 2. 해당 좌석이 예매되면 예매된 자리의 상태가 바뀌고, VIP 회원의 할인 쿠폰이 적용됨        1. 프로그램을 종료하면 프로그램이 종료되면서 안내 메시지가 출력됨 |

의견

기말고사 이후에도 과제가 있어 씁쓸한 마음이 있었으나, 기한은 넉넉하게 주셨기 때문에 한 편으로는 씁쓸한 마음을 덜 수 있었습니다. 이 프로그램을 만들면서 아직 배울 것이 많다는 걸 수도 없이 느꼈으나, 반대로 이만큼 해낸 자신을 돌아보니 대견하다는 생각도 들었습니다. 배울 것이 많은 만큼 미숙한 점도 많겠지만, 이를 계기로 포기하지 않고 꾸준히 노력하다 보면 머지 않아 정상을 찍을 수 있지 않을까 라는 기대도 해봅니다. 한 학기 동안 고생 많으셨습니다. 코로나 조심하시고 날이 부쩍 추워졌으니 건강 잘 챙기시길 바랍니다. 감사합니다.