

기능상세 설계서

B711128 이상욱

1. 프로그램 사용 매뉴얼 및 화면 설계서

해당 프로그램은 코로나 시대 사회적 거리두기 현황에 맞추어 예매할 수 있게 설계된 영화 예매 프로그램입니다.

프로그램 실행 시 우선 메뉴 선택창이 있는데

```
=====
홍익극장에 오신 것을 환영합니다!
=====

아래 번호를 입력해 주세요.

1. 영화 예매하기
2. 영화 예매 취소
3. 관리자 설정(좌석 통계 확인하기)
4. 프로그램 종료
=====
```

1. 영화예매 하기 2. 영화 예매 취소 3. 관리자 설정(좌석 통계) 4. 프로그램 종료

로 나뉩니다. 해당 사항 중 사용자가 원하는 메뉴를 실행하기 위해서는 해당 번호를 입력해야 합니다.

(1).영화 예매하기

```
성함을 입력해주세요 : Lee
성별(번호)을 선택해주세요 : 1.남자 2.여자
1
휴대전화 번호를 입력해 주세요 .(공백x)
01071113333
```

1번을 누르면 영화를 예매할 수 있는 화면으로 넘어갑니다. 그러면 사용자의 이름과 성별 휴대전화 번호를 입력 순차적으로 입력하면 됩니다. 이때 성별은 1.남자, 2.여자로 사용자 성별의 번호를 입력하면 되고 전화번호는 공백 없이 입력하면 됩니다.

```
=====
현재 상영중인 영화는
1.로마의 휴일
2.티파니에서 아침을
3.카사블랑카
4.바람과 함께 사라지다
=====
선택하실 영화의 번호를 입력해주세요: 1
```

개인정보를 입력하면 상영중인 영화목록이 나옵니다. 보고 싶은 영화의 번호를 입력하면 됩니다.

```
=====
현재 상영중인 영화는
1.로마의 휴일
2.티파니에서 아침을
3.카사블랑카
4.바람과 함께 사라지다
=====
선택하실 영화의 번호를 입력해주세요: 1

해당 영화의 상영관은 1관, 2관 입니다.

1관 상영시간
1. 10 : 30 ~ 12: 30.
2. 15 : 30 ~ 17: 30.
3. 21 : 30 ~ 23: 30.
2관 상영시간
1. 10 : 00 ~ 12: 00.
2. 15 : 30 ~ 17: 30.
3. 20 : 30 ~ 22: 30.
원하시는 상영관의 숫자를 입력해 주세요 : 1
```

다음으로는 상영관을 입력하면 되는데 해당 영화관은 총 4개의 영화를 8개의 상영관에서 상영하고 첫번째 영화는 1,2관 두번째 영화는 3,4관 세번째 영화는 5,6관 네번째 영화는 7,8관에서만 각각 상영됩니다. 따라서 하나의 영화 선택 후 두 가지 상영관 중 사용자가 원하는 상영관을 그대로 입력하시면 됩니다. EX) 1관 -> 1

원하시는 상영시간의 번호를 입력해 주세요: 1

각각의 상영관은 다른 상영시간을 가지므로 같은 영화라도 다양한 시간대를 선택할 수 있습니다. 마찬가지로 원하시는 상영시간의 번호를 입력하시면 됩니다.

```
좌석 현황
x 1 2 3 4 5
A o o o o o
B o o o o o
C o o o o o
D o o o o o
E o o o o o

원하시는 좌석 번호를 입력해주세요. EX)A열 3번 자리 -> a3 or A3
```

그 후 해당 상영관의 상영대에 맞는 좌석 현황이 노출됩니다. 해당 영화관의 상영관 1~8 관 모두 5x5개의 좌석 수를 가지고 세로는 A,B,C,D,E 가로는 1,2,3,4,5로 해당 좌석에 자리가 있어서 예매 가능하면 "o"로 표시되고 이미 다른 사람이 예매했거나 사회적 거리두기에 따라 예매 불가능한 좌석이면 "x"로 표시됩니다.

원하시는 좌석 번호를 입력해주세요. EX)A열 3번 자리 -> a3 or A3
a3
선택하신 좌석은 A열3번 자리 입니다.

좌석 현황

x	1	2	3	4	5
A	o	x	x	x	o
B	o	o	x	o	o
C	o	o	o	o	o
D	o	o	o	o	o
E	o	o	o	o	o

좌석을 선택하면 해당 좌석 외 앞뒤 좌우 좌석모두 예매 불가능하므로 "x"표시로 바뀌는 것을 확인할 수 있습니다.

영화 가격은 10000원 입니다.
잔여금은 10000원 입니다.

포인트 적립을 위해 휴대전화 번호를 입력해주세요 (공백x)

영화 예매 가격은 10000원이고 모든 사용자는 기본으로 20000원을 가지고 있다고 했습니다. 따라서 잔여금은 10000원이 됩니다. 그 다음 포인트 적립을 위해 반드시 본인이 입력한 휴대전화 번호 그대로를 입력해야 합니다.

포인트 적립을 위해 휴대전화 번호를 입력해주세요 (공백x)
01071113333
Lee님의 포인트는 100P 입니다.

그러면 다음과 같이 포인트가 적립되는 것을 확인할 수 있습니다. 포인트 적립까지 완료하면 영화 예매 메뉴는 완료됩니다.

(2).영화 예매 취소

```
=====
아래 번호를 입력해 주세요.
1. 영화 예매하기
2. 영화 예매 취소
3. 관리자 설정(좌석 통계 확인하기)
4. 프로그램 종료
=====
번호: 2
성함을 입력해 주세요:Lee
휴대전화 번호를 입력해 주세요:01071113333
영화 예매를 취소하시겠습니까?
1. Yes 2. No
```

메뉴 선택창에서 2번을 누르면 영화 예매 취소 기능이 실행되는데 예매 취소시 우선 취소하고 싶은 사용자의 이름과 전화번호를 입력해야 합니다. 해당 정보는 맨처음 입력 시 입력했던 정보와 일치해야합니다.

10000원 환불됩니다.
잔여금은 20000원 입니다.
포인트 차감을 위해 휴대전화 번호를 입력해주세요.(공백x)

그 후 예매 취소를 뜻하는 숫자 1을 입력하면 사용자에게 금액이 환불되고 적립했던 포인트 차감을 위해 다시한번 휴대전화 번호를 입력받습니다.

```
포인트 차감을 위해 휴대전화 번호를 입력해주세요.(공백x)
01071113333
Lee님의 포인트는 0P 입니다.

좌석 현황
x 1 2 3 4 5
A x x o o o
B x o o o o
C o o o o o
D o o o o o
E o o o o o

선택하신 좌석은 A열1번 자리 입니다.
해당 좌석의 예매가 취소 되었습니다.

좌석 현황
x 1 2 3 4 5
A o o o o o
B o o o o o
C o o o o o
D o o o o o
E o o o o o
```

전화번호를 입력하면 사용자의 이름과 전화번호로 예매한 자리를 찾아 자동으로 취소되게 하고 그 상영관의 상영대의 좌석현황이 변경됩니다.

(3).관리자 설정(좌석 통계확인)

관리자 설정은 좌석통계 기능을 확인하기 위한 메뉴로 숫자 3입력을 통해 실행할 수 있습니다.

```
=====
아래 번호를 입력해 주세요.
1. 영화 예매하기
2. 영화 예매 취소
3. 관리자 설정(좌석 통계 확인하기)
4. 프로그램 종료
=====
번호: 3
관리자 설정 입니다.
원하시는 통계 데이터에 해당하는 번호를 입력해주세요.
1. 1일차
2. 2일차
3. 3일차
4. 4일차
5. 5일차
6. 6일차
7. 7일차(당일 까지의 통계합)
8. 좌석별 세부정보
9. 관리자 설정 종료
번호:
```

1번부터 6번까지는 지난 6일간 영화관을 다녀간 사람들의 통계를 확인할 수 있습니다.

(※ 1일부터 6일 까지의 사용자 데이터는 하드 코딩 했으며 함수 정의 부에서 인적데이터를 변경 하면 그에 따라 통계 값도 변하게 설계했습니다.)

7번은 1일차부터 7일차까지의 통계로 아무 예매도 없이 누르면 6일차까지의 통계만 나오게 됩니다.

```
번호: 7
=====
1관에서 가장 많이 예매된 좌석 Top1은 총7석으로 E열5번 자리 입니다.
1관에서 가장 많이 예매된 좌석 Top2은 총6석으로 B열4번 자리 입니다.
1관에서 가장 많이 예매된 좌석 Top3은 총5석으로 A열1번 자리 입니다.
=====
2관에서 가장 많이 예매된 좌석 Top1은 총10석으로 A열5번 자리 입니다.
2관에서 가장 많이 예매된 좌석 Top2은 총7석으로 E열1번 자리 입니다.
2관에서 가장 많이 예매된 좌석 Top3은 총6석으로 D열2번 자리 입니다.
=====
3관에서 가장 많이 예매된 좌석 Top1은 총13석으로 E열5번 자리 입니다.
3관에서 가장 많이 예매된 좌석 Top2은 총6석으로 B열3번 자리 입니다.
3관에서 가장 많이 예매된 좌석 Top3은 총5석으로 C열5번 자리 입니다.
=====
4관에서 가장 많이 예매된 좌석 Top1은 총11석으로 E열1번 자리 입니다.
4관에서 가장 많이 예매된 좌석 Top2은 총10석으로 A열1번 자리 입니다.
4관에서 가장 많이 예매된 좌석 Top3은 총6석으로 C열1번 자리 입니다.
=====
```

통계는

가장 많이 예매된 좌석을 상영관마다 Top3

```
영화별 가장 많이 예매된 좌석
1. 로마의 휴일
가장 많이 예매된 좌석 Top1은 총14석으로 C열1번 자리 입니다.
가장 많이 예매된 좌석 Top2은 총10석으로 A열1번 자리 입니다.
가장 많이 예매된 좌석 Top3은 총8석으로 E열5번 자리 입니다.
2. 티파니에서 아침을
가장 많이 예매된 좌석 Top1은 총27석으로 E열1번 자리 입니다.
가장 많이 예매된 좌석 Top2은 총23석으로 A열1번 자리 입니다.
가장 많이 예매된 좌석 Top3은 총20석으로 A열5번 자리 입니다.
3. 가사불량가
가장 많이 예매된 좌석 Top1은 총45석으로 E열5번 자리 입니다.
가장 많이 예매된 좌석 Top2은 총42석으로 E열1번 자리 입니다.
가장 많이 예매된 좌석 Top3은 총36석으로 A열1번 자리 입니다.
4. 바람과 함께 사라지다
가장 많이 예매된 좌석 Top1은 총65석으로 E열1번 자리 입니다.
가장 많이 예매된 좌석 Top2은 총62석으로 E열5번 자리 입니다.
가장 많이 예매된 좌석 Top3은 총49석으로 A열1번 자리 입니다.
=====
```

영화별로 가장 많이 예매된 좌석을 Top3

(※각각의 영화의 상영관이 고정되어 있어 해당 영화를 상영하는 상영관들의 합계로 통계 내었습니다.)

```

남자고객이 가장 많이 예매한 좌석
=====
가장 많이 예매된 좌석 Top1은 총46석으로 A열5번 자리 입니다.
가장 많이 예매된 좌석 Top2은 총44석으로 A열1번 자리 입니다.
가장 많이 예매된 좌석 Top3은 총29석으로 E열1번 자리 입니다.
=====

```

```

여자고객이 가장 많이 예매한 좌석
=====
가장 많이 예매된 좌석 Top1은 총57석으로 E열5번 자리 입니다.
가장 많이 예매된 좌석 Top2은 총36석으로 E열1번 자리 입니다.
가장 많이 예매된 좌석 Top3은 총30석으로 D열2번 자리 입니다.
=====

```

성별에 따라 많이 예매한 좌석 Top3

순차적으로 나오게 하였습니다.

```

관리자 설정 입니다.
원하시는 통계 데이터에 해당하는 번호를 입력해주세요.
1. 1일차
2. 2일차
3. 3일차
4. 4일차
5. 5일차
6. 6일차
7. 7일차(당일까지의 통계합)
8. 좌석별 세부정보
9. 관리자 설정 종료

번호: 8

```

```

=====
1. 영화별 예매횟수
2. 남녀 예매 비율
3. 상영대별 예매 횟수
=====

번호:

```

8번을 입력하면 좌석별 세부정보를 확인할 수 있는데 세부 정보에 해당하는 숫자를 입력하면 됩니다.

```

=====
1. 영화별 예매횟수
2. 남녀 예매 비율
3. 상영대별 예매 횟수
=====

번호: 1
원하시는 좌석 번호를 입력해주세요. EX)A열 3번 자리 -> a3 or A3
a5

A열5번째 자리의 영화별 예매횟수
로마의 휴일 10회
티파니에서 아침을 10회
카사블랑카 11회
바람과 함께 사라지다. 15회

```

영화별 예매횟수는 좌석 번호를 입력하면 1~8관까지의 해당 좌석에 해당하는 영화별 예매횟수가

나오게 됩니다.

```
=====
1. 영화별 예매횟수
2. 남녀 예매 비율
3. 상영대별 예매 횟수
=====

번호: 2

원하시는 상영관을 입력해 주세요(1관~8관): 1
1. 10 : 30 ~ 12: 30.
2. 15 : 30 ~ 17: 30.
3. 21 : 30 ~ 23: 30.
원하시는 시간대를 입력해 주세요(1타임~3타임): 1

원하시는 좌석 번호를 입력해주세요. EX)A열 3번 자리 -> a3 or A3
a3
1관 1타임 A열3번째 자리의 남녀별 예매횟수

남자 0회
여자 5회
```

남녀 예매 비율은 숫자 2를 입력한 다음 상영관, 상영대를 입력한 후 좌석 번호를 입력합니다.

```
=====
1. 영화별 예매횟수
2. 남녀 예매 비율
3. 상영대별 예매 횟수
=====

번호: 3

원하시는 상영관을 입력해 주세요(1관~8관): 1

원하시는 좌석 번호를 입력해주세요. EX)A열 3번 자리 -> a3 or A3
a3

1관 A열3번째 자리의 상영대별 예매횟수
1. 10 : 30 ~ 12: 30. 타임5회
2. 15 : 30 ~ 17: 30. 타임0회
3. 21 : 30 ~ 23: 30. 타임0회
```

상영대별 예매 횟수는 상영관과 좌석번호를 입력하면 해당 상영관에 해당하는 상영대별로 예매 횟수가 나오게 됩니다.

2. 클래스 정의서 및 인스턴스간 상호작용

(1) Class Person

Person 클래스는 영화를 예매하는 사용자를 객체로 산정해 설계한 클래스입니다.

따라서 사용자 개개인으로써 가지는 이름, 휴대전화 번호, 성별 변수를 선언하였습니다.

또한 사용자가 프로그램을 통해 실행하는 영화 예매 및 예매 취소와 관련된 행동을 구현

하기 위해 Movie라는 클래스의 객체를 선언 하였고 포인트 적립을 하고 적립을 취소하는 행동을 구현하기 위해 Point라는 클래스의 객체를 선언하였습니다. 또한 Person의 객체끼리 공유할 수 있는 상영관마다 좌석이 필요하여 Seat클래스의 객체를 배열이자 static인 상태로 선언하였습니다. static Seat seat[8][3] 8:상영관 개수 3: 상영대 개수

상단의 변수 중 개인의 정보를 나타내는 이름, 휴대전화, 성별 , Point 객체는 private으로 좌석과 영화선택은 public으로 설계하였습니다.

멤버함수는

```
bool Check_info(string name, string phone_num)
```

main함수에서 예매 취소 시 입력받은 매개변수(이름, 휴대전화번호)를 통해 예매된 좌석에 저장된 개인 정보와 비교하기 위한 함수입니다.

```
void Set_hard(string name, int gender, string phone_num, int movie_choice, int theater, int time_idx);
```

6일간의 데이터(개인정보 및 영화, 상영관 선택)를 입력 받기 위해서 7일째와 다르게 설계할 필요가 있었습니다. 따라서 하드코딩용 개인의 정보 및 영화 선택 데이터를 입력받기 위한 함수를 설계했습니다. 하드 코딩된 내용은 위 매개변수 순서에 따라 얼마든지 다르게 입력할 수 있습니다.

```
void personal_information();
```

7일째 사용자에게 직접 개인정보를 입력 받기 위한 함수입니다.

```
void Reservation();
```

7일째 사용자가 예매하기 위한 기능들을 종합적으로 포함하는 함수입니다. void personal_information(), Movie 객체 내 void Movie_choice() , Seat객체 내 void Seat_output()//좌석 현황 출력 함수 void Seat_input(string name, int gender, string phone_num) 좌석에 개인정보를 전달하기 위한 함수, void PayandSave() 결제 및 포인트 적립을 위한 함수를 포함합니다.

```
void Reservation_hard(char x, char y);
```

6일간의 데이터를 예매하기 위한 예매 기능 함수로 하드 코딩 된 좌석 번호를 입력 받아 Seat객체의 void Seat_input_hard(char x, char y, int gender, string phone_num)에 넘기게 됩니다.

```
void PayandSave();
```

결제 및 포인트 적립을 위한 함수로 void Payment(int price) void Save(string name) 각각 결제와 포인트 적립기능을 하는 함수를 포함합니다.

```
void Reservation_cancel()
```

영화예매를 취소하기 위한 함수로 가격을 환불받는 void Refund() 함수와 개인이 등록한 정보를 삭제하는 Movie 객체 내 void Delete() 함수를 포함합니다.

```
void Refund();
```

결제 환불과 적립된 포인트 차감을 위한 함수로 void Cancel_payment(int price) 함수와 void Resave(string name) 함수를 포함합니다.

(2) Class Movie

Movie클래스는 Person 클래스를 통해 생성된 객체 내부에 선언되어 사용자가 입력한 영화(영화관) 관련 정보들을 객체화 한 것입니다.

멤버 변수로 영화 선택 값을 받는 int movie_choice 상영관 선택 값을 받는 int theater 상영관 상영시간 정보를 저장한 string time[8][3] 상영시간 선택 값을 받는 int time_idx 그리고 영화가격을 저장한 int price가 있습니다. 위 멤버 변수는 모두 private으로 설계했습니다.

멤버 함수는 public으로 설계하고

```
void Movie_choice()
```

영화선택 기능을 하기위해 설계된 함수입니다. 멤버 변수 movie_choice에 사용자로부터 입력 받은 영화 번호를 넘겨받아 해당 영화에 맞는 상영관과 상영시간 또한 변수를 통해 입력 받는 기능을 합니다.

```
void Delete();
```

Person 객체가 영화 예매 취소 시 Person 객체 내 멤버 Movie m1을 통해 호출되는 함수 입니다. 취소 시 사용자와 관련된 영화정보를 삭제하기 위한 기능을 합니다.

```
friend Seat;
```

```
friend Person;
```

Person 클래스와 Seat 클래스에서 생성된 객체가 Movie 클래스 내 멤버에 접근할수 있게 하기 위해 friend선언을 하였습니다.

(3) Class Seat_info

사용자가 예매할 때 좌석을 선택하는데 그 정보들을 좌석의 관점에서 객체화 한 것입니

다.

사용자 정보인 int gender, string name , string phone_num 사용자가 선택한 좌석의 열과 번호를 나타내는 int seat_row ,int seat-col이 멤버 변수이고

멤버함수로는

```
void Update_seatInfo(string name, int gender, string phone_num, int seat_row, int seat_col)
```

좌석에 해당 사용자 정보를 입력받는 기능을 하는 함수입니다.

```
void Delete_seatInfo(string name, int gender, string phone_num, int seat_row, int seat_col)
```

예매 취소시 해당 사용자 정보를 삭제하는 기능을 하는 함수입니다.

```
bool Compare(string name, string phone_num)
```

예매 취소 시 넘겨 받은 이름과 휴대전화 번호 값을 해당 좌석에 저장된 번호와 비교하는 기능을 하는 함수입니다. 실제 사용자를 구별하는 것은 휴대전화 번호이기 때문에 사용되는 것은 string phone_num 매개변수 이지만 사실감을 위하여 이름값 또한 입력받아 넘겨받을 수 있도록 설계하였습니다.

```
friend Seat
```

```
friend Cnt
```

Cnt 와 Seat 클래스에서 해당 클래스의 멤버에 접근할수 있도록 friend선언을 하였습니다.

(4) class Seat

각각의 상영관마다 있는 좌석 그 자체를 객체화 하였습니다. 멤버변수로는 Seat_info의 객체를 배열로 선언하였습니다. 사용자로부터 입력받은 좌석 번호를 저장하기 위한 char input_seat[2] 배열을 선언하였습니다. Int temp_x , int temp_y는 char형으로 입력받은 좌석 번호를 int형으로 변환하여 저장하기 위한 변수입니다.

좌석현황을 콘솔 어플리케이션에 표시하기 위해 char seat_ui[6][6] 배열을 선언하였습니다. 통계 시 활용하기 위한 int cnt_seat[6][6] int gender_seat[2][6][6] 배열을 선언하였습니다.

생성자 함수로는 멤버 배열인 seat_ui[6][6]을 초기화 하여 좌석 현황을 표시하기 위한 문자값들을 대입합니다. Void seat_output() 함수는 seat_ui 배열을 통해 콘솔 화면에 배열값

을 출력하는 기능을 합니다. Void seat_check()함수는 사회적 거리두기를 적용하여 사용자가 선택한 좌석 앞뒤 좌우 모두 예매 불가하도록 seat_ui배열값들을 변경하는 기능을 합니다. Void seat_check_for_candle()은 사용자가 예매 취소시 해당 좌석 외 사회적 거리두기로 예매 불가한 좌석 모두 예매 가능하게 seat_ui배열값을 변경하는 기능을 합니다.

Void Seat_input(string name, int gender, string phone_num) 함수는 이름,성별,휴대전화번호를 매개변수로 넘겨받음과 동시에 좌석을 사용자로부터 입력받는 기능을 하도록 설계하였습니다. Void Seat_input_hard()는 직접 입력 받는 것이 아닌 하드 코딩하여 좌석 정보와 사용자 정보를 매치하기 위한 기능을 하도록 설계하였습니다. Void Reset()함수는 7일간 좌석 정보를 입력하고 변경하기 위해 Seat클래스 객체 내 멤버 변수를 초기값으로 초기화하는 기능을 하도록 설계하였습니다. Void seat_candle(string name, int gender, string phone_num)은 예매 취소 시 좌석과 해당 사용자의 정보들을 삭제하고 초기화하는 기능을 지원하는 함수입니다.

(5) Class Point

사용자가 결제 및 포인트 적립 및 결제 취소 포인트 적립 취소 시 오가는 정보들을 객체화 하기 위해 설계한 클래스입니다.

멤버 변수로는 int charge_money , charge_point가 있습니다.

멤버함수로는 위 변수들을 초기화하는 생성자 함수

가격을 매개변수로 받아 결제기능을 하는 void Payment(int price)함수

포인트를 적립하는 기능을 하는 void Save(string name) 함수

결제 취소 기능을 하는 void Cacle_payment(int price) 함수

포인트 적립 취소 기능을 하는 void Cacle_save(string name) 함수

를 멤버 함수로 가진다.

(6) Class Cnt

통계 계산을 위한 클래스입니다.

Seat* s1 객체 내 seat_ui[6][6] 배열을 사용하기 위해 선언된 객체포인터입니다. 동적할당을 통해 생성하도록 하였습니다.

static s2[8][3] static으로 선언하여 앞서 Person 내 선언된 Seat클래스 seat객체에 저장된 좌석 정보들을 입력 받는 객체 배열입니다.

Movie* m을 동적할당 하여 객체 포인터로 선언한다. Movie 클래스 내 string time배열에 접근하기 위함으로 메모리 부담이 적은 동적할당을 선택하였습니다.

멤버 함수로는

~Cnt() 소멸자 함수를 통해 동적할당하여 생성된 객체포인터 s1 m을 delete하기 위함입니다.

Void Sum_seats()

좌석별로 통계 데이터를 받아 계산하기 위한 함수입니다.

Void ChartToInt()

통계 메뉴에서 좌석별 통계 정보를 확인하고자 할 때 해당 좌석을 입력받아 정수 값으로 변환하는 기능을 하는 함수입니다. Out_sum()함수에서 호출됩니다.

Void Out_sum()

좌석별 세부 정보에서 영화별, 남녀별, 좌석별 예매 정보를 출력하는 함수입니다.

Void Set_gender() void Cnt_gender() 성별로 가장 많이 예매된 좌석을 위해 setting 하고 계산하는 함수들입니다.

Void Set_movie() void Cnt_movie() 영화별로 가장 많이 예매된 좌석을 위해 setting 하고 계산하는 함수들입니다.

Void Set_most() void Cnt_most() 가장 많이 예매된 좌석을 위해 setting 하고 계산하는 함수들입니다.

Out_gender , Out_most , Out_movie 계산된 값들을 출력하기 위한 함수입니다.

(7) Class Execution

Cnt클래스 객체들을 c1,c2 ... c8까지 선언하여(각각 1일차, 2일차...7일차,1~7일차까지의 통계 합) 각각 날짜의 통계를 입력받을 객체 역할을 한다.

멤버함수로는 void Day1To7_Execution() main함수에서 호출되어 c1~c7 객체 내 멤버함수들을 호출하여 통계 기능을 호출 및 출력하는 기능을 하는 함수입니다. 7일차는 직접 입력 받아야 하므로 관련된 인터페이스를 출력하는 기능도 합니다.

(8) 기타 함수

void Reset_Seat()

static Seat seat[8][3] 배열을 하루 지날 때 마다 초기값으로 초기화하는 기능을 수행합니다.

void Day1()

day1의 영화관 방문자들을 하드코딩한 함수입니다. Day1()함수 내 Person객체가 동적으로 생산되어 각각 정보를 입력받고 정보를 변경하면 통계값도 변경됩니다.

void Day2()

day2의 영화관 방문자들을 하드코딩한 함수입니다. Day2()함수 내 Person객체가 동적으로 생산되어 각각 정보를 입력받고 정보를 변경하면 통계값도 변경됩니다.

void Day3()

day3의 영화관 방문자들을 하드코딩한 함수입니다. Day3()함수 내 Person객체가 동적으로 생산되어 각각 정보를 입력받고 정보를 변경하면 통계값도 변경됩니다.

void Day4()

day4의 영화관 방문자들을 하드코딩한 함수입니다. Day4()함수 내 Person객체가 동적으로 생산되어 각각 정보를 입력받고 정보를 변경하면 통계값도 변경됩니다.

void Day5()

day5의 영화관 방문자들을 하드코딩한 함수입니다. Day5()함수 내 Person객체가 동적으로 생산되어 각각 정보를 입력받고 정보를 변경하면 통계값도 변경됩니다.

void Day6();

day6의 영화관 방문자들을 하드코딩한 함수입니다. Day6()함수 내 Person객체가 동적으로 생산되어 각각 정보를 입력받고 정보를 변경하면 통계값도 변경됩니다.

