
영화예매 프로그램

Python 개인 프로그래밍 과제

과목명	창의소프트웨어입문(F115-4)		
교수님	고종원 교수님		
제출일	2021.12.20		
학과/학번/이름	e비즈니스학과	201823871	박지현
E-mail	jiheon788@ajou.ac.kr		



아주대학교
AJOU UNIVERSITY

목차

1. 서론: 문제확인하고 작업환경	- 3 -
1.1. 요구사항과 추가 기능	- 3 -
1.2. 개발환경	- 4 -
1.3. 프로젝트 파일 개요	- 5 -
2. 본론: 소스코드와 주석	- 6 -
2.1. <code>__init__.py</code>	- 6 -
2.2. <code>theater.py</code>	- 6 -
2.3. <code>reservation.py</code>	- 7 -
2.4. <code>payment.py</code>	- 11 -
2.5. <code>admin.py</code>	- 13 -
2.6. <code>main.py</code>	- 16 -
3. 결론: 실행결과 및 개선점	- 24 -
3.1. 실행결과	- 24 -
3.2. 개선점	- 34 -
3.3. 표와 그림 인덱스	- 36 -

1. 서론: 문제확인하고 작업환경

1.1. 요구사항과 추가 기능

문제: 배열을 이용해서 간단한 극장예약 시스템을 구현하라.
아주 작은 극장이어서 좌석이 10X10 이며,
예약이 끝난 좌석은 1로, 예약이 안된 좌석은 0으로 나타난다.

문제의 요구사항은 아래 표와 같으며, 요구사항 이외의 추가기능을 구현하였다. 추가기능 또한 표에서 확인할 수 있다.

요구사항	10 x 10 의 좌석배열(예약 1, 예약 x 0)	○
	영화제목 선택	○
	성인/청소년 몇 장 구매할 것인지	○
	상영관 및 상영시간 선택	○
	좌석 예약	○
	결제금액 계산	○
	최종 선택한 내용 및 결제금액 결과 출력	○
추가기능	거스름돈 계산	○
	티켓 출력	○
	티켓 출력을 위한 인증코드 발행	○
	고객 DB 관리	○
	관리자채널 로그인, 고객인증코드 일치검사	○

Table 1 문제 요구사항과 추가기능 리스트

구현한 프로그램의 소스코드와 주석은 '2장 본문: 소스코드와 주석' 파트에서 텍스트로 기재하였으며, 실행결과는 '3장 결론: 실행결과 및 한계' 파트에서 확인할 수 있다.

1.2. 개발환경

Python 3.9.2
PyCharm Community Edition 2021.1.3 x64
Windows 10
64비트 운영 체제

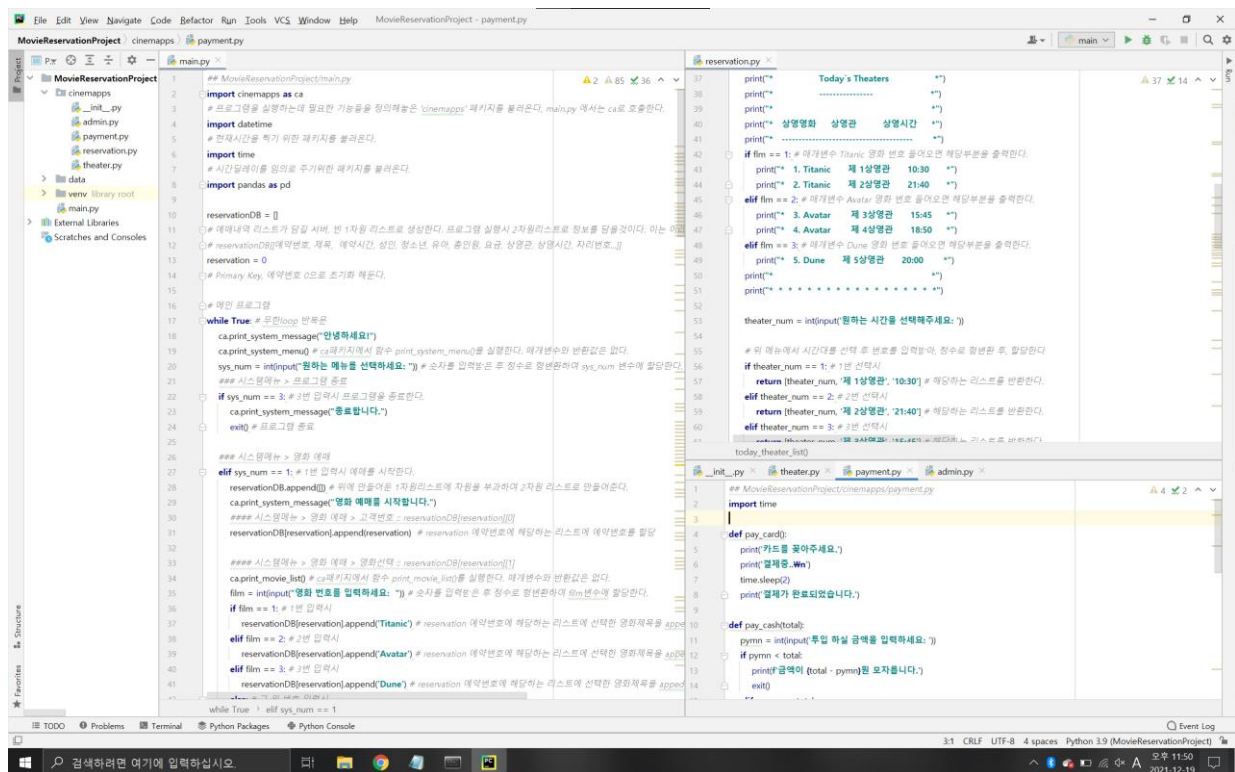


Figure 1 프로젝트 개발 화면

1.3. 프로젝트 파일 개요

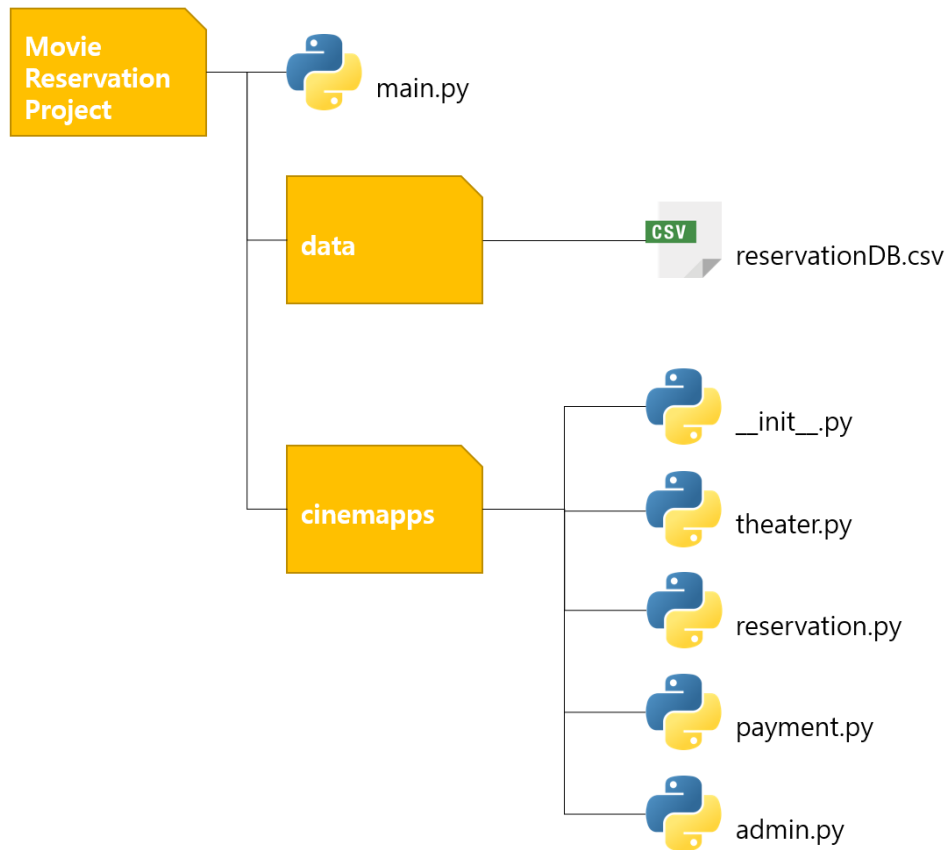


Figure 2 MovieReservationProject 개요

main.py는 본 프로그램을 실행하는 메인 파일이다.

`main.py` 코드의 가독성을 높이기 위한 간소화 & 코드의 재사용성을 높이기 위한 모듈화를 하기위해 **cinemapps**(‘시네맵스’) 패키지를 만들었다. 재사용이 가능한 모든 함수와 몇몇 변수, 상수들은 각 기능에 맞는 모듈에 모아두었다.

시네맵스의 내부에는 초기화를 위한 **__init__.py**, 상영관별 좌석 정보가 담긴 이중리스트들을 모아 놓은 **theater.py**, 예약을 위해 필요한 함수들을 모아 놓은 **reservation.py**, 결제에 필요한 기능을 모아 놓은 **payment.py**, 로그인 및 본인인증코드 발급, DB관리를 위한 함수 등 관리자 레벨에 필요한 기능을 담은 **admin.py**가 있다.

`data`폴더에는 고객들의 예매내역이 담긴 데이터베이스 csv파일로 존재한다. 메인 파일에서 관리자채널로 이동하여 당일 모든 고객의 예매내역을 날짜별로 저장 가능하다.

각 모듈의 내부 정의된 함수는 '제 2장 본문: 소스코드와 주석'에서 각 파트별로 앞서 간단한 기능을 소개할 예정이다. 함수의 알고리즘 및 소스코드에서 라인 별로 주석을 달아 자세히 설명을 기재하였다.

2. 본문: 소스코드와 주석

2.1. __init__.py

```
## MovieReservationProject/cinemapps/__init__.py
# cinemapps 패키지의 초기화파일
# main.py 에서 cinemapps 패키지를 import 하면 제일 먼저 이 파일을 열어 아래 코드를 읽는다.
# from A.a import *: A 폴더의 a 파일로부터 모두 import 한다.
from cinemapps.reservation import *
# cinemapps 패키지의 reservation 모듈로부터 전부 import 한다.
from cinemapps.theater import *
# cinemapps 패키지의 theater 모듈로부터 전부 import 한다.
from cinemapps.admin import *
# cinemapps 패키지의 admin 모듈로부터 전부 import 한다.
from cinemapps.payment import *
# cinemapps 패키지의 payment 모듈로부터 전부 import 한다.
```

Path: MovieReservationProject/cinemapps/__init__.py

2.2. theater.py

각각의 상영 영화와 시간이 다른 5개의 상영관의 이중리스트가 담겨있다.

```
## MovieReservationProject/cinemapps/theater.py
## 오늘의 영화별 상영관 좌석 리스트 초기화 10 x 10

Titanic_H1 = [[0 for col in range(10)] for row in range(10)]
## 이중 for 문을 사용하여 10x10 이중리스트를 0으로 초기화한다.
Titanic_H2 = [[0 for col in range(10)] for row in range(10)]
## 이중 for 문을 사용하여 10x10 이중리스트를 0으로 초기화한다.
Avatar_H3 = [[0 for col in range(10)] for row in range(10)]
## 이중 for 문을 사용하여 10x10 이중리스트를 0으로 초기화한다.
Avatar_H4 = [[0 for col in range(10)] for row in range(10)]
## 이중 for 문을 사용하여 10x10 이중리스트를 0으로 초기화한다.
```

```
Dune_H5 = [[0 for col in range(10)] for row in range(10)]
## 이중 for 문을 사용하여 10x10 이중리스트를 0으로 초기화한다.
```

Path: MovieReservationProject/cinemapps/theater.py

2.3. reservation.py

print_system_menu(): 시스템 메뉴를 출력해준다.

print_movie_list(): 영화리스트를 출력해준다.

print_today_theater_n_choice(flm): 선택한 영화를 넣으면 영화에 맞는 상영관과 시간대 리스트를 출력하고 선택한 후 값들을 반환해준다.

print_seats(movie): 영화좌석정보가 담긴 이중리스트를 넘기면 형식에 맞추어 출력해준다. 좌석정보를 확인할 수 있다.

choice_seats(i, movie): 여러 장의 좌석 선택을 대비해 몇 번째 반복인지 i와 선택한 상영관을 매개변수로 받아 좌석을 선택한다. 이미 예약된 자리거나, 없는 좌석일시 경고문을 출력한다.

print_ticket(rsrvtDB, rsrvt): 데이터베이스와 출력할 예약번호를 매개변수로 받아 티켓을 지정된 형식으로 출력해준다. 지류티켓이 나오는 효과를 주기위해 time패키지를 사용해 임의의 딜레이를 주었다.

```
## MovieReservationProject/cinemapps/reservation.py
from pprint import pprint
# 리스트의 출력을 위한 pprint 함수 impotr 한다.
import time
# 임의의 딜레이를 위하여 time 패키지 import 한다.

## 함수 정의 > 시스템메뉴 출력
def print_system_menu(): # 매개변수 x, 반환값 x
    print("* * * * *")
    print("          MENU          ")
    print("          -----          ")
    print("          *")
    print("1.  영화 예매          ")
    print("2.  티켓 출력          ")
    print("3.  종료              *")
```

```

print("** ----- **")
print("** 0. 관리자 **")
print("** **")
print("** * * * * * * * * * * * * * * * * * * * * **")

```

함수 정의 > 영화리스트 출력

```

def print_movie_list(): # 매개변수 x, 반환값 x
    print("** * * * * * * * * * * * * * * * * * * * * **")
    print("**           Movie List           **")
    print("**           -----           **")
    print("** **")
    print("** 1. Titanic **")
    print("** 2. Avatar **")
    print("** 3. Dune **")
    print("** **")
    print("** * * * * * * * * * * * * * * * * * * * * **")

```

함수 정의 > 상영관 정보

```

def print_today_theater_n_choice(flm):
    # 매개변수 선택한영화 번호, 반환값: [상영관키넘버, 상영관문자열, 상영시간문자열]
    print("** * * * * * * * * * * * * * * * * * * * * **")
    print("**           Today's Theaters           **")
    print("**           -----           **")
    print("** **")
    print("** 상영영화   상영관   상영시간   **")
    print("** ----- **")
    if flm == 1: # 매개변수 Titanic 영화 번호 들어오면 해당부분을 출력한다.
        print("** 1. Titanic   제 1 상영관   10:30   **")
        print("** 2. Titanic   제 2 상영관   21:40   **")
    elif flm == 2: # 매개변수 Avatar 영화 번호 들어오면 해당부분을 출력한다.
        print("** 3. Avatar   제 3 상영관   15:45   **")
        print("** 4. Avatar   제 4 상영관   18:50   **")
    elif flm == 3: # 매개변수 Dune 영화 번호 들어오면 해당부분을 출력한다.
        print("** 5. Dune   제 5 상영관   20:00   **")
    print("** **")
    print("** * * * * * * * * * * * * * * * * * * * * **")

```

theater_num = int(input('원하는 시간을 선택해주세요: '))

번호를 입력받아 정수로 형변환한다.

위 메뉴에서 시간대를 선택 후 번호를 입력받아, 정수로 형변환 후, 할당한다


```

if theater_num == 1:
    # 1 번 선택시
    return [theater_num, '제 1 상영관', '10:30']
    # 해당하는 리스트를 반환한다.
elif theater_num == 2:
    # 2 번 선택시
    return [theater_num, '제 2 상영관', '21:40']
    # 해당하는 리스트를 반환한다.
elif theater_num == 3:
    # 3 번 선택시
    return [theater_num, '제 3 상영관', '15:45']
    # 해당하는 리스트를 반환한다.
elif theater_num == 4:
    # 4 번 선택시
    return [theater_num, '제 4 상영관', '18:50']
    # 해당하는 리스트를 반환한다.
elif theater_num == 5:
    # 5 번 선택시
    return [theater_num, '제 5 상영관', '20:00']
    # 해당하는 리스트를 반환한다.

## 함수 정의 > 좌석현황 출력
def print_seats(movie):
    # 선택한 상영관의 좌석정보가 담긴 이중리스트를 매개변수로 전달받아 출력해준다.
    print("=====")
    print("          SCREEN          ")
    print("=====")
    pprint(movie)
    #이중 for 문을 사용하지 않아도 이중리스트를 출력해준다.
    print("=====| Gate |==")

## 함수 정의 > 자리 선택
def choice_seats(i, movie):
    #매개변수: i 와 선택한 상영관의 좌석정보
    print_seats(movie)
    # 위 정의된 함수 출력, 좌석현황을 보여준다. 예약 1, 빈자리 0
    x, y = input(f'{i + 1}번째 자리를 입력하세요(10 10): ').split()
    # i 는 0 부터이기에 +1 하여 순서를 알려준다.
    # 숫자 2 개를 간격을 두어 입력 받은 후 split 하여 각각 x, y 두가지 변수에 할당한다.
    x = int(x) # 변수를 정수로 변환한 뒤 다시 저장
    y = int(y) # 변수를 정수로 변환한 뒤 다시 저장

```

```

if x > 10 or y > 10 :
    #범위 밖 번호를 입력시 안내문 출력 후 종료
    print('없는 좌석입니다.')
    exit()
if movie[x - 1][y - 1] == 1:
    # 리스트의 인덱스도 0 부터 시작한다. 고로 입력받은 번호 - 1 해준다
    # 선택한 자리가 이미 1 이면, 해당좌석은 예약이 되어 있는 자리다.
    # 예약불가 안내메시지 출력 후, 종료.
    print('이미 예약된 좌석입니다.')
    exit()
movie[x - 1][y - 1] = 1
# 위 두가지 조건문에 걸리지 않으면 선택한 자리를 1 로 바꾸어준다.
return x, y # 선택된 좌석좌표를 반환한다.

# 함수 정의 > 티켓 및 예매내역 출력
def print_ticket(rsrvtDB, rsrvt):
    # 함수 정의, 매개변수로 예약내역이 모두 담긴 이중리스트, 예약번호를 받는다.
    print('출력중..Wn')
    time.sleep(2)
    # 넘어가는데 2 초 임의로 딜레이를 준다.
    print("|", "".center(33, 'V'), "|")
    # 33 칸 가운데 정렬 나머지는 V로채운다.
    print("|", "Ticket".center(59, ' '), "|")
    # 59 칸 가운데 정렬 나머진 공백으로 채운다.
    print("| ", "".center(58, ' '), " |")
    # 공백으로 58 칸 가운데정렬
    time.sleep(0.5)
    # 결제가 넘어가는데 0.5 초 임의로 딜레이를 준다.
    print("| ".ljust(6), f"{rsrvtDB[rsrvt][2]}".ljust(23), " ".rjust(20), "|".rjust(6), sep="")
    # 왼쪽정렬 6 칸, 왼쪽정렬 20 칸, 오른쪽정렬 20 칸, 오른쪽정렬 6 칸, 구분자없음
    print("| ", "".center(50, '-'), " |")
    # 가운데 정렬 50 칸 하이픈으로 채운다.
    time.sleep(0.5)
    # 넘어가는데 0.5 초 임의로 딜레이를 준다.
    print("| ".ljust(6), f"Movie  :".ljust(20), f"{rsrvtDB[rsrvt][1]}".rjust(20), "|".rjust(14), sep="")
    # 왼쪽정렬 6 칸, 왼쪽정렬 20 칸, 오른쪽정렬 20 칸, 오른쪽정렬 14 칸, 구분자없음
    print("| ".ljust(6), f"Num   :".ljust(20), f"{rsrvtDB[rsrvt][6]}".rjust(20), "|".rjust(15), sep="")
    # 왼쪽정렬 6 칸, 왼쪽정렬 20 칸, 오른쪽정렬 20 칸, 오른쪽정렬 15 칸, 구분자없음
    print("| ".ljust(6), f"Area   :".ljust(20), f"{rsrvtDB[rsrvt][8]}".rjust(20), "|".rjust(8), sep="")
    # 왼쪽정렬 6 칸, 왼쪽정렬 20 칸, 오른쪽정렬 20 칸, 오른쪽정렬 8 칸, 구분자없음
    time.sleep(0.5)

```

```

# 넘어가는데 0.5 초 임의로 딜레이를 준다.
print("| ".ljust(6), f"Time      :".ljust(20), f"{rsrvtDB[rsrvt][9]}".rjust(20), "|".rjust(14), sep="")
# 왼쪽정렬 6 칸, 왼쪽정렬 20 칸, 오른쪽정렬 20 칸, 오른쪽정렬 14 칸, 구분자없음
seat_num = 11
# 좌석정보는 11 열부터 나온다.
for i in range(0, rsrvtDB[rsrvt][6]):
    # 0 부터 인원수 - 1 까지 반복한다.
    time.sleep(0.5)
    # 넘어가는데 0.5 초 임의로 딜레이를 준다.
    print("| ".ljust(6), f"Seat      :".ljust(20), f"({rsrvtDB[rsrvt][seat_num]})".rjust(20), "|".rjust(17), sep="")
    # 왼쪽정렬 6 칸, 왼쪽정렬 20 칸, 오른쪽정렬 20 칸, 오른쪽정렬 17 칸, 구분자없음
    # 좌석정보 reservationDB[reservation][11] 부터 인원수 만큼 순차적으로 출력한다
    seat_num += 1
    # seat_num 을 한자리 올려서 반복, 다음 좌석의 열번호이다.
print("| ", "".center(50, '-'), " |")
# 가운데 정렬 50 칸 하이픈으로 채운다.
print("| ", "".center(50, '-'), " |")
# 가운데 정렬 50 칸 하이픈으로 채운다.
print("| ".ljust(6), f"Total      ".ljust(20), f"W {rsrvtDB[rsrvt][7]}".rjust(19), "|".rjust(14), sep="")
# 왼쪽정렬 6 칸, 왼쪽정렬 20 칸, 오른쪽정렬 19 칸, 오른쪽정렬 14 칸, 구분자없음
print("| ", "".center(50, '-'), " |")
# 가운데 정렬 50 칸 하이픈으로 채운다.
time.sleep(0.5)
# 넘어가는데 0.5 초 임의로 딜레이를 준다.
print("| ", "".center(58, ' '), " |")
# 공백으로 58 칸 가운데정렬
print("|", "".center(33, 'V'), "|")
# 33 칸 가운데 정렬 나머지는 V로채운다.
print('WnWn')
# 두줄띄운다.
time.sleep(1)
# 넘어가는데 1 초 임의로 딜레이를 준다.

```

Path: MovieReservationProject/cinemapps/reservation.py

2.4. payment.py

pay_card(): 카드 결제를 위한 함수이다. 카드 결제중 이라는 디테일을 위해 time패키지를 이용해 임의의 딜레이를 주었다.

pay_cash(ttl_amnt): 결제액을 매개변수로 받으면, 함수내부에서 투입금액을 입력받아, 결제를 완료해준다. 투입금액이 모자를 때, 같을 때 그리고 초과됐을 때의 거스름돈 지폐 수 계산 기능을 넣었다.

```
## MovieReservationProject/cinemapps/payment.py
import time
# 시간딜레이를 임의로 주기위한 패키지 import.

def pay_card():
    # 함수 정의, 매개변수 x
    print('카드를 꽂아주세요.')
    print('결제중..Wn')
    time.sleep(2)
    # 결제가 넘어가는데 2 초 임의로 딜레이를 준다.
    print('결제가 완료되었습니다.')

def pay_cash(ttl_amnt):
    # 함수정의, 매개변수: 결제할 총 금액
    pymn = int(input('투입 하실 금액을 입력하세요: '))
    # 투입금액을 입력받아 정수로 형변환한다,
    if pymn < ttl_amnt:
        # 투입금액이 결제 금액보다 모자르면
        print(f'금액이 {ttl_amnt - pymn}원 모자릅니다.')
        # 모자를 금액 출력
        exit()
        # 프로그램 종료
    elif pymn == ttl_amnt:
        # 투입금액과 결제 금액이 같다면
        print('결제가 완료되었습니다.')
        # 결제완료
    else:
        # 투입금액이 결제금액보다 크다면 거스름돈 출력
        change = pymn - ttl_amnt
        # 잔돈 = 투입금액 - 결제금액
        krw50000 = int(change / 50000)
        # 50000 으로 나눈 몫 만큼 지급
        change = change % 50000
        # 50000 으로 나눈 나머지를 다시 할당
        krw10000 = int(change / 10000)
        # 10000 으로 나눈 몫 만큼 장수지급
        change = change % 10000
```

```

# 10000 으로 나눈 나머지를 다시 할당.
krw5000 = int(change / 5000)
# 5000 으로 나눈 몫 만큼 장수지급
change = change % 5000
# 5000 으로 나눈 나머지를 다시 할당.
krw1000 = int(change / 1000)
# 1000 으로 나눈 몫 만큼 장수지급
change = change % 1000
# 1000 으로 나눈 나머지를 다시 할당.

print('--거스름돈--'.center(20))
# 20 칸 가운데 정렬 출력
print(f"50000 원권 : {krw50000}장")
# 해당범주에 맞는 지폐 장수 출력
print(f"10000 원권 : {krw10000}장")
# 해당범주에 맞는 지폐 장수 출력
print(f"5000 원권 : {krw5000}장")
# 해당범주에 맞는 지폐 장수 출력
print(f"1000 원권 : {krw1000}장")
# 해당범주에 맞는 지폐 장수 출력
print(f"동전:      {change}원")
# 남은 동전 출력
print('결제가 완료되었습니다.')

```

Path: MovieReservationProject/cinemapps/payment.py

2.5. admin.py

ADMIN_PASSWORD: 관리자 비밀번호를 상수로 설정했다.

print_admin_channel(): 관리자 채널의 메뉴를 보여준다.

print_system_message(message): 시스템 안내, 경고 메시지를 출력하기 위해 문자열을 받아 특정 형식으로 출력해준다.

generate_code(): 무작위의 6자리 인증코드를 반환해준다. 티켓출력시 본인인증을 위해 필요하다.

customer_login(athntCode): 티켓출력을 위해 본인인증용 인증코드와 입력한 인증코드

일치를 검사하고 맞으면 True를 반환해준다. 3번의 시도횟수를 초과하면 False를 반환한다.

admin_login(): 관리자 채널에 들어가기 위해 관리자 비밀번호와 입력한 비밀번호 일치를 검사하고 맞으면 True를 반환해준다. 3번의 시도횟수를 초과하면 False를 반환한다.

```
## MovieReservationProject/cinemapps/admin.py
from random import *
# 랜덤패키지를 모두 import 한다.

#####
ADMIN_PASSWORD = 12345678
#####
## #####관리자 비밀번호
## #####상수로 설정해둔다.

def print_admin_channel():
    # 함수정의, 매개변수 x, 반환값 x
    print("* * * * *")
    print("          Admin Channel          ")
    print("          -----          ")
    print("                                     *")
    print("1. DB 출력                                     *")
    print("2. DB 저장                                     *")
    print("                                     *")
    print("* * * * *")

def print_system_message(message):
    # 함수정의, 매개변수로 메시지를 받는다.
    print(f"Wn=====WnSystem:
{message}Wn=====Wn")
    #메시지를 특정형식으로 출력한다. 시스템 메시지 출력용

def generate_code():
    # 반환값만 있는 함수 정의
    alphabet = "abcdefghijklmnopqrstuvwxyz0123456789"
    # 알파벳과 숫자 문자열
    code = ""
    # 빈문자열 정의
    for i in range(6):
        # 0 부터 5 까지 총 6 번 반복
```

```

index = randrange(len(alphabet))
# 0 부터 알파벳변수의 길이까지의 랜덤한정수를 뽑아 인덱스로 만든다.
code += alphabet[index]
# code 에 해당인덱스를 가지 문자를 할당한다.
return code
# code 를 반환한다.

def customer_login(athntCode):
# 인증코드를 매개변수로 받는다.
for i in range(3):
# 0 ~ 2 까지, 세번 반복한다.
inputCode = input(f"인증코드를 입력하세요({3 - i}/3): ")
# 인증코드를 입력받아 문자열 그대로 할당, 남은 로그인 시도 횟수를 보여준다.
if inputCode != athntCode:
# 입력한 코드가 다르다면 안내문 출력한다.
print("인증실패. ")
if (3 - i) == 1:
# i 가 2 가 되었을때 = 기회를 세번 모두 썼을 때
return False
# False 반환 > 로그인 실패
else:
# 아직 기회가 남아있으면
continue
# 다시 반복
elif inputCode == athntCode:
#입력한 코드가 맞으면
return True
# True 반환 > 인증완료

def admin_login():
# 함수 정의, 로그인
for i in range(3):
# 0 ~ 2 까지, 세번 반복한다.
inputPW = int(input(f"패스워드를 입력하세요({3 - i}/3): "))
# 패스워드를 입력받고 정수로 형변환 후 할당, 남은 로그인 시도 횟수를 보여준다.
if inputPW != ADMIN_PASSWORD:
# 입력한 비밀번호가 다르다면 안내문 출력한다.
print("틀렸습니다. ")
if (3 - i) == 1:
# i 가 2 가 되었을때 = 기회를 세번 모두 썼을 때
return False

```

```

        # False 반환 > 로그인 실패
    else:
        # 아직 기회가 남아있으면
        continue
        # 다시 반복
    elif inputPW == ADMIN_PASSWORD:
        #입력한 비밀번호가 맞으면
        return True
        # True 반환 > 로그인 성공

```

Path: MovieReservationProject/cinemapps/admin.py

2.6. main.py

```

## MovieReservationProject/main.py
import cinemapps as ca
# 프로그램을 실행하는데 필요한 기능들을 정의해놓은 'cinemapps' 패키지를 불러온다, main.py 에서는
ca 로 호출한다.
import datetime
# 현재시간을 찍기 위한 패키지 import
import time
# 시간딜레이를 임의로 주기위한 패키지 import.
import pandas as pd
# 데이터베이스를 관리 하기위한 외부 패키지 import, main.py 에서는 pd 로 호출한다.

reservationDB = []
# 예매내역 리스트를 빈 1 차원 리스트로 생성한다. 프로그램 실행시 2 차원리스트로 정보를 담을것이다.
이는 아래와 같다.
# reservationDB[[예약번호, 제목, 예약시간, 성인, 청소년, 유아, 총인원, 요금, 상영관, 상영시간, 인증코드,
자리번호...]]
reservation = 0
# Primary Key, 예약번호: 0 으로 초기화 해둔다.

# 메인 프로그램
while True:
    # 무한 loop 반복문
    ca.print_system_message("안녕하세요!")
    # ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할 메시지를 넣으면
    함수에서 지정해둔 형식으로 출력해준다.

```



```

ca.print_system_menu()
# ca 패키지에서 함수 print_system_menu()을 실행한다. 매개변수와 반환값은 없다.
sys_num = int(input("원하는 메뉴를 선택하세요: "))
# 숫자를 입력받은 후 정수로 형변환하여 sys_num 변수에 할당한다.

### 시스템메뉴 > 프로그램 종료
if sys_num == 3:
    # 3 번 입력시 프로그램을 종료한다.
    ca.print_system_message("종료합니다.")
    # ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할 메시지를 넣으면
    함수에서 지정해둔 형식으로 출력해준다.
    exit()
    # 프로그램 종료

### 시스템메뉴 > 영화 예매
elif sys_num == 1:
    # 1 번 입력시 예매를 시작한다.
    reservationDB.append([])
    # 위에 만들어둔 1 차원리스트에 차원을 부과하여 2 차원 리스트로 만들어준다.
    ca.print_system_message("영화 예매를 시작합니다.")
    # ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할 메시지를 넣으면
    함수에서 지정해둔 형식으로 출력해준다.

##### 시스템메뉴 > 영화 예매 > 고객번호 :: reservationDB[reservation][0]
reservationDB[reservation].append(reservation)
# reservationDB[reservation]에 예약번호를 할당

##### 시스템메뉴 > 영화 예매 > 영화선택 :: reservationDB[reservation][1]
ca.print_movie_list()
# ca 패키지에서 함수 print_movie_list()를 실행한다. 매개변수와 반환값은 없다.
film = int(input("영화 번호를 입력하세요: "))
# 숫자를 입력받은 후 정수로 형변환하여 film 변수에 할당한다.
if film == 1:
    # 1 번 입력시
    reservationDB[reservation].append('Titanic')
    # reservationDB[reservation]에 선택한 영화제목을 apped
elif film == 2:
    # 2 번 입력시
    reservationDB[reservation].append('Avatar')
    # reservationDB[reservation]에 선택한 영화제목을 apped
elif film == 3:

```

```

# 3 번 입력시
reservationDB[reservation].append('Dune')
# reservationDB[reservation]에 선택한 영화제목을 append
else:
    # 그 외 번호 입력시
    ca.print_system_message("잘못된 입력입니다.")
    # ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할 메시지를
    # 넣으면 함수에서 지정해둔 형식으로 출력해준다.
    exit()
    # 프로그램 종료

#### 시스템메뉴 > 영화 예매 > 예매시각(현재시간) :: reservationDB[reservation][2]
now = datetime.datetime.now()
# datetime 패키지의 now() 함수를 사용하여 현재시간을 now 에 할당한다.
reservationDB[reservation].append(now.strftime('%Y-%m-%d %H:%M:%S'))
# reservationDB[reservation]에 현재시간을 주어진 포맷의 문자열로 할당한다. '년-월-일 시:분:초'

#### 시스템메뉴 > 영화 예매 > 성인,청소년,유아 인원 수 입력
adult, youth, infant = input('성인, 청소년, 유아 티켓 수를 입력하세요(n n n): ').split()
# 숫자 세개를 간격을 두어 입력 받은 후 split 하여 각각 세가지 변수에 할당한다.
adult = int(adult)
# 변수를 정수로 변환한 뒤 다시 저장
youth = int(youth)
# 변수를 정수로 변환한 뒤 다시 저장
infant = int(infant)
# 변수를 정수로 변환한 뒤 다시 저장
reservationDB[reservation].append(adult)
#### 성인 :: reservationDB[reservation][3]
# reservationDB[reservation]에 성인 수를 append 한다.
reservationDB[reservation].append(youth)
#### 청소년 :: reservationDB[reservation][4]
# reservationDB[reservation]에 청소년 수를 append 한다.
reservationDB[reservation].append(infant)
#### 유아 :: reservationDB[reservation][5]
# reservationDB[reservation]에 유아 수를 append 한다.

#### 시스템메뉴 > 영화 예매 > 총 인원수:: reservationDB[reservation][6]
total_people = adult + youth + infant
# 성인 + 청소년 + 유아 수를 더해 총 인원을 구한다
reservationDB[reservation].append(total_people)
# reservationDB[reservation]에 총인원 수를 append 한다.

```

```

#### 시스템메뉴 > 영화 예매 > 요금 :: reservationDB[reservation][7]
total_amount = 10000 * adult + 8000 * youth + 0 * infant
# 성인 10000 원, 청소년 8000 원, 유아 0 원으로 계산하여 총 요금을 할당한다.
reservationDB[reservation].append(total_amount)
# reservationDB[reservation]에 총 요금을 append 한다.

#### 시스템메뉴 > 영화 예매 > 상영관 & 상영시간
selected_theater = ca.print_today_theater_n_choice(film)
# ca 패키지의 print_today_theater_n_choice(film) 함수에 매개변수로 영화번호를 입력하여,
상영관을 선택한 후 리스트로 값을 반환받는다.
# selected_theater = [상영관키넘버, 상영관문자열, 상영시간문자열]
reservationDB[reservation].append(selected_theater[1])
#### 상영관 :: reservationDB[reservation][8]
# reservationDB[reservation]에 상영관을 append 한다.
reservationDB[reservation].append(selected_theater[2])
#### 상영시간 :: reservationDB[reservation][9]
# reservationDB[reservation]에 상영시간을 append 한다.

#### 시스템메뉴 > 영화 예매 > 인증코드 :: reservationDB[reservation][10]
authenticationCode = ca.generate_code()
# ca 패키지의 generate_code 를 사용하여 인증코드를 받는다.
reservationDB[reservation].append(authenticationCode)
# reservationDB[reservation]에 상영시간을 append 한다.

#### 시스템메뉴 > 영화 예매 > 좌석예약 :: reservationDB[reservation][11~]
for i in range(0, total_people):
    # 0 부터 인원수-1 까지 반복하여 인원 수 만큼 티켓을 예매한다.
    if selected_theater[0] == 1:
        # 위에서 입력받은 상영관키넘버 1 번일 때,
        x, y = ca.choice_seats(i, ca.Titanic_H1)
        # ca 패키지의 Titanic_H1 상영관 과 i 를 매개변수로 choice_seats(i, 상영관) 함수에
        입력하여 x, y 값을 반환 받는다.
    elif selected_theater[0] == 2:
        # 위에서 입력받은 상영관키넘버 2 번일 때,
        x, y = ca.choice_seats(i, ca.Titanic_H2)
        # ca 패키지의 Titanic_H2 상영관 과 i 를 매개변수로 choice_seats(i, 상영관) 함수에
        입력하여 x, y 값을 반환 받는다.
    elif selected_theater[0] == 3:
        # 위에서 입력받은 상영관키넘버 3 번일 때,
        x, y = ca.choice_seats(i, ca.Avatar_H3)

```

```

        # ca 패키지의 Avatar_H3 상영관 과 i 를 매개변수로 choice_seats(i, 상영관) 함수에
        입력하여 x, y 값을 반환 받는다.
        elif selected_theater[0] == 4:
            # 위에서 입력받은 상영관키넘버 4 번일 때,
            x, y = ca.choice_seats(i, ca.Avatar_H4)
            # ca 패키지의 Avatar_H4 상영관 과 i 를 매개변수로 choice_seats(i, 상영관) 함수에
            입력하여 x, y 값을 반환 받는다.
        elif selected_theater[0] == 5:
            # 위에서 입력받은 상영관키넘버 5 번일 때,
            x, y = ca.choice_seats(i, ca.Dune_H5)
            # ca 패키지의 Dune_H5 상영관 과 i 를 매개변수로 choice_seats(i, 상영관) 함수에
            입력하여 x, y 값을 반환 받는다.
        reservationDB[reservation].append(f'{x}, {y}')
        # reservationDB[reservation]에 좌석좌표 x, y 를 append 한다.

#### 시스템메뉴 > 영화 예매 > 예매내역출력 & 결제
ca.print_system_message("결제창으로 넘어갑니다.")
# ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할 메시지를 넣으면
함수에서 지정해둔 형식으로 출력해준다.
time.sleep(0.5)
# 결제창으로 넘어가는데 0.5 초 임의로 딜레이를 준다.
print(f"=====예매내역 확인=====")
print(f" 영화                {reservationDB[reservation][1]}")
# reservationDB[reservation][1] 내용을 출력한다.
print(f" 상영관                {reservationDB[reservation][8]}")
# reservationDB[reservation][8] 내용을 출력한다.
print(f" 상영시간            {now.strftime('%Y 년 %m 월 %d 일')} {reservationDB[reservation][9]}")
# 위에서 datetime 패키지를 통해 받은 오늘의 날짜를 지정형식의 문자열로 가져와
reservationDB[reservation][9] 내용과 함께 출력한다.
seat_num = 11
# 좌석정보는 11 열부터 나온다.
for i in range(0, reservationDB[reservation][6]):
    # 0 부터 인원수 - 1 까지 반복한다.
    print(f" 좌석                {reservationDB[reservation][seat_num]}")
    # 좌석정보 reservationDB[reservation][11] 부터 인원수 만큼 순차적으로 출력한다
    seat_num += 1
    # seat_num 을 한자리 올려서 반복, 다음 좌석의 열번호이다.
print(f"=====")
print(f" 성인                {reservationDB[reservation][3]}명")
# reservationDB[reservation][3] 내용을 출력한다.
print(f" 청소년            {reservationDB[reservation][4]}명")

```

```

# reservationDB[reservation][4] 내용을 출력한다.
print(f" 유아          {reservationDB[reservation][5]}명")
# reservationDB[reservation][5] 내용을 출력한다.
print(f"=====")
print(f" Total          WW {reservationDB[reservation][7]}Wn")
# 총 요금 출력한다, 원화 표시를 위해 W를 그대로 출력하려면 WW두번쓴다.
pymn_methods = int(input("1. 카드    2. 현금: "))
# 결제방식을 입력받아 정수로 변환 후 할당한다.

#### 시스템메뉴 > 영화 예매 > 결제 > 카드
if pymn_methods == 1:
    # 카드결제시,
    ca.pay_card()
    # ca 패키지의 pay_card()실행.

#### 시스템메뉴 > 영화 예매 > 결제 > 현금
elif pymn_methods == 2:
    # 현금 결제시,
    ca.pay_cash(total_amount)
    # ca 패키지의 pay_cash()실행.

ca.print_system_message(f"예약완료!Wn 예약번호: {reservation}Wn 인증코드:
{authenticationCode}Wn 티켓 출력시 예약번호와 인증코드가 필요합니다.")
# ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할 메시지를 넣으면
함수에서 지정해둔 형식으로 출력해준다.
# 예약번호와 인증코드도 같이 보여준다. 추후 티켓출력시 인증을 위해 필요하다
reservation += 1
# 예약번호를 한자리 올리고 예매끝. 다시 while 문 처음으로 돌아간다.

### 시스템메뉴 > 티켓 출력
elif sys_num == 2:
    #sys_num 2 번 티켓출력 선택
    reservation_num = int(input("예약번호를 입력하세요: "))
    #예약을 하고 받은 예약번호를 입력한다.
    if ca.customer_login(reservationDB[reservation_num][10]) == True:
        # ca 패키지의 customer_login()함수에 매개변수로 예약번호에 해당하는 인증코드를 넣어
        함수내부에서 코드를 입력받아 일치하는지 확인한다.
        # 일치하다면 True 를 반환받아 인증성공
        ca.print_system_message("인증성공!")
        # ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할 메시지를
        넣으면 함수에서 지정해둔 형식으로 출력해준다.

```

```

ca.print_ticket(reservationDB, reservation_num)
# ca 패키지의 print_ticket() 함수, 예약리스트가 담긴 reservationDB 와 출력하려는
예약번호를 매개변수로 넘긴다.

else:
    # False 를 반환받았다면 코드를 세번 틀린것. 인증실패
    ca.print_system_message("인증실패!")
    # ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할 메시지를
    넣으면 함수에서 지정해둔 형식으로 출력해준다.
    exit()
    #프로그램종료

### 시스템메뉴 > 관리자 채널
elif sys_num == 0:
    #sys_num 0 번 관리자탭 선택
    ca.print_system_message("관리자채널 입니다.")
    # ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할 메시지를 넣으면
    함수에서 지정해둔 형식으로 출력해준다.
    if ca.admin_login() == True:
        #ca 패키지의 admin_login() 함수 호출, 반환값이 True 이면, 로그인성공
        ca.print_system_message("로그인 성공!")
        # ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할 메시지를
        넣으면 함수에서 지정해둔 형식으로 출력해준다.
        ca.print_admin_channel()
        # ca 패키지의 print_admin_channel() 함수 메뉴를 보여준다.
        admin_num = int(input("번호를 입력하세요: "))
        # 원하는 메뉴를 선택한다. 숫자를 문자로 입력받아 정수로 형변환한다.
        if admin_num == 1:
            # 1 번 선택
            ReservationDatabase = pd.DataFrame(reservationDB)
            # 만들어진 이중리스트를 판다스 데이터프레임으로 변환하여 ReservationDatabase 에
            저장한다.

            print(ReservationDatabase)
            # 데이터 프레임을 출력한다.

        elif admin_num == 2:
            # 2 번 선택
            timestamp = now.strftime('%Y%m%d')
            # 변수 now 에 할당되어있는 현재시간을 주어진 포맷의 문자열로 할당한다. '년월일'
            ReservationDatabase = pd.DataFrame(reservationDB)
            # 만들어진 이중리스트를 판다스 데이터프레임으로 변환하여 ReservationDatabase 에
            저장한다.

```

```

ReservationDatabase.to_csv(f'data/reservationDB{timestamp}.csv',index=False,header=False)
    # 데이터프레임 ReservationDatabase 를 지정된 위치에 csv 파일로만든다. 뒤에 년월일을
    # 붙임, 인덱스 없고, 열이름도 지정하지 않는다.
    ca.print_system_message("DB 저장완료")
    # ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할
    # 메시지를 넣으면 함수에서 지정해둔 형식으로 출력해준다.
else:
    # 로그인 실패시 접속불가
    ca.print_system_message("로그인 실패!")
    # ca 패키지의 print_system_message("message")를 실행한다. 매개변수로 출력할 메시지를
    # 넣으면 함수에서 지정해둔 형식으로 출력해준다.
break
#프로그램종료

```

Path: MovieReservationProject/main.py

3. 결론: 실행결과 및 개선점

3.1. 실행결과

```
=====
System: 안녕하세요!
=====

* * * * *
*           MENU           *
*   -----               *
*
* 1. 영화 예매             *
* 2. 티켓 출력             *
* 3. 종료                 *
*   -----               *
* 0. 관리자                 *
*
* * * * *

원하는 메뉴를 선택하세요: 1

=====
System: 영화 예매를 시작합니다.
=====

* * * * *
*           Movie List      *
*   -----               *
*
* 1. Titanic               *
* 2. Avatar                *
* 3. Dune                  *
*
* * * * *

영화 번호를 입력하세요: 1
성인, 청소년, 유아 티켓 수를 입력하세요(n n n): 2 1 0
```

Figure 3 첫번째 고객의 예약화면 1


```

* * * * *
*           Today`s Theaters           *
*           -----                     *
*
* 상영영화   상영관   상영시간   *
* -----
* 1. Titanic   제 1상영관   10:30   *
* 2. Titanic   제 2상영관   21:40   *
*
* * * * *
원하는 시간을 선택해주세요: 1
=====
SCREEN
=====
[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
=====| Gate |=
1번째 자리를 입력하세요(10 10): 1 1

```

Figure 4 첫번째 고객의 예약화면 2 - 상영관 선택 및 좌석 선택

```

=====
                SCREEN
=====
[[1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
=====| Gate |=
2번째 자리를 입력하세요(10 10): 1 2
=====
                SCREEN
=====
[[1, 1, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
=====| Gate |=
3번째 자리를 입력하세요(10 10): 1 3

```

Figure 5 첫번째 고객의 예약화면 3 - 좌석선택

```

=====예매내역 확인=====
영화      Titanic
상영관     제 1상영관
상영시간   2021년 12월 22일 10:30
좌석       1, 1
좌석       1, 2
좌석       1, 3

=====

성인       2명
청소년     1명
유아       0명

=====

Total      ₩ 28000

1. 카드   2. 현금: 1
카드를 꽂아주세요.
결제중..

결제가 완료되었습니다.

=====
System: 예약완료!
예약번호: 0
인증코드: m7t0i0
티켓 출력시 예약번호와 인증코드가 필요합니다.
=====

```

Figure 6 첫번째 고객의 예약화면 4 - 예매 내역 확인, 카드결제 및 예약완료

```

* * * * *
*           MENU           *
*           -----        *
*                               *
* 1.  영화 예매           *
* 2.  티켓 출력           *
* 3.  종료                 *
* -----                *
* 0.  관리자               *
*                               *
* * * * *

원하는 메뉴를 선택하세요: 2
예약번호를 입력하세요: 0
인증코드를 입력하세요(3/3): 2352fe
인증실패.
인증코드를 입력하세요(2/3): fe23
인증실패.
인증코드를 입력하세요(1/3): m7t0i0

=====
System: 인증성공!
=====

```

Figure 7 첫번째 고객의 티켓출력 1 - 예약번호, 인증코드를 통한 본인인증

출력중..

~~~~~	
Ticket	
2021-12-22 15:55:04	
-----	
Movie :	Titanic
Num :	3
Area :	제 1상영관
Time :	10:30
Seat :	(1, 1)
Seat :	(1, 2)
Seat :	(1, 3)
-----	
-----	
Total	₩ 28000
-----	
~~~~~	

Figure 8 첫번째 고객의 티켓출력 2 - 티켓출력

```

=====예매내역 확인=====
영화      Titanic
상영관     제 1상영관
상영시간   2021년 12월 22일 10:30
좌석       5, 5

=====

성인       1명
청소년     0명
유아       0명

=====

Total      ₩ 10000

1. 카드   2. 현금: 2
투입 하실 금액을 입력하세요: 37500
    --거스름돈--
50000원권 : 0장
10000원권 : 2장
5000원권  : 1장
1000원권  : 2장
동전:      500원
결제가 완료되었습니다.

=====
System: 예약완료!
예약번호: 1
인증코드: x8z87u
티켓 출력시 예약번호와 인증코드가 필요합니다.
=====

```

Figure 9 두번째 고객의 예약화면 - 현금결제의 거스름돈 계산 및 예약완료

Ticket	
2021-12-22 15:55:35	

Movie :	Titanic
Num :	1
Area :	제 1상영관
Time :	10:30
Seat :	(5, 5)

Total	₩ 10000

- 31 -

```

=====
System: 안녕하세요!
=====

* * * * *
*           MENU           *
*           -----        *
*                               *
* 1.  영화 예매           *
* 2.  티켓 출력           *
* 3.  종료                 *
*           -----        *
* 0.  관리자               *
*                               *
* * * * *

원하는 메뉴를 선택하세요: 0

=====
System: 관리자채널 입니다.
=====

패스워드를 입력하세요(3/3): 3124
틀렸습니다.
패스워드를 입력하세요(2/3): 12345678

=====
System: 로그인 성공!
=====

```

Figure 11 관리자 채널 1 - 패스워드를 통한 관리자 로그인


```

* * * * *
*           Admin Channel           *
*           -----                 *
*                                     *
* 1. DB 출력                       *
* 2. DB 저장                       *
*                                     *
* * * * *

번호를 입력하세요: 1
0      1          2 3 4 ... 9    10 11 12 13
0 0 Titanic 2021-12-22 15:55:04 2 1 ... 10:30 m7t0i0 1, 1 1, 2 1, 3
1 1 Titanic 2021-12-22 15:55:35 1 0 ... 10:30 x8z87u 5, 5 None None

[2 rows x 14 columns]

```

Figure 12 관리자 채널 2 - 관리자 메뉴와 저장된 DB출력

```

* * * * *
*           Admin Channel           *
*           -----                 *
*                                     *
* 1. DB 출력                       *
* 2. DB 저장                       *
*                                     *
* * * * *

번호를 입력하세요: 2

=====
System: DB저장완료
=====

```

Figure 13 관리자 채널 3 - 관리자 메뉴와 DB저장

MovieReservationProject > data			
이름	수정된 날짜	유형	크기
reservationDB20211219.csv	2021-12-19 오후 10:50	Microsoft Excel 실효...	1KB
reservationDB20211222.csv	2021-12-22 오후 3:57	Microsoft Excel 실효...	1KB

Figure 14 지정된 루트에 날짜별로 저장된 DB

reservationDB20211222.csv - Excel														
<div>파일 홈 삽입 페이지 레이아웃 수식 데이터 검토 보기 도움말 어떤 작업을 원하십니까?</div> <div> <div>잘라내기 붙여넣기 서식 복사 클립보드 글꼴 맞춤 표시 형식</div> <div> <div>읽은 고딕 11</div> <div>가 가</div> <div>가 가 가</div> <div>가 가</div> <div>가 가</div> <div>가 가</div> <div>가 가</div> <div>가 가</div> <div>가 가</div> <div>가 가</div> <div>가 가</div> <div>가 가</div> <div>가 가</div> <div>가 가</div> <div>가 가</div> </div> </div>														
D5														
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	0	Titanic	2021-12-22 15:55	2	1	0	3	28000	??1?곡뽕뽕	10:30 m7t0i0	1, 1	1, 2	1, 3	
2	1	Titanic	2021-12-22 15:55	1	0	0	1	10000	??1?곡뽕뽕	10:30 x8z87u	5, 5			
3														
4														
5														

Figure 15 2021.12.22일자 예약내역이 담긴 csv파일

3.2. 개선점

문제점 1: 현재 프로그램에서 지정범위를 넘어서는 숫자 입력 등의 오류가 발생하면 모두 잘못된 입력을 알리는 안내문 출력과 함께 `exit()`문을 사용하였다. 이는 바로 프로그램을 종료시키며 DB저장을 미리하지않았다면 모든 저장내역이 소멸된다. 원래는 `break`문이나 `continue`문 을 사용하여 프로그램을 종료시키지 않아야 하지만 이를 사용시 Primary Key인 리스트의 첫번째 열에 들어가는 예약번호(reservationDB[reservation][0])가 꼬이는 문제가 발생하여 우선 프로그램을 종료시키는 방향으로 프로그램을 만들었다.

c언어에서의 `goto`문과 같은 명령어가 있다면 내가 지정한 위치로 돌려 다시 반복을 시킬 수 있다. 지금까지 찾아본 바로는 파이썬에서는 그러한 기능을 하는 명령어를 찾지

못하였다.

문제점 2: 파이썬의 기본 인코딩 방식은 utf-8을 사용한다. 이는 한글을 저장하기에 적합한 인코딩 방식으로 알고 있으나 DB저장시 한글이 깨지는 현상이 발생한다.

추후 개선할점: 현재 관리자 채널에서는 DB를 출력하거나 지정된 위치에 csv파일로 저장하는 기능뿐이지만, 후에 f-string을 사용하여 원하는 과거 날짜의 데이터를 가져오는 기능도 추가할 것이다.

```
past_date = input("가져오실 과거 날짜를 입력하세요(yyyymmdd): ")
```

```
pastDB = pd.read_csv(f'data/reservationDB{past_date}')
```

이러한 코드로 과거데이터를 가져와 열람할 수 있을것같으나, 열람이외에 활용방안이 떠오르지않아 작성하지 않았다.

3.3. 표와 그림 인덱스

Table 1 문제 요구사항과 추가기능 리스트	- 3 -
Figure 1 프로젝트 개발 화면	- 4 -
Figure 2 MovieReservationProject 개요	- 5 -
Figure 3 첫번째 고객의 예약화면 1	- 24 -
Figure 4 첫번째 고객의 예약화면 2 - 상연관 선택 및 좌석 선택	- 25 -
Figure 5 첫번째 고객의 예약화면 3 - 좌석선택	- 26 -
Figure 6 첫번째 고객의 예약화면 4 - 예매 내역 확인, 카드결제 및 예약완료	- 27 -
Figure 7 첫번째 고객의 티켓출력 1 - 예약번호, 인증코드를 통한 본인인증	- 28 -
Figure 8 첫번째 고객의 티켓출력 2 - 티켓출력	- 29 -
Figure 9 두번째 고객의 예약화면 - 현금결제의 거스름돈 계산 및 예약완료	- 30 -
Figure 10 두번째 고객의 티켓출력 - 예약번호, 인증코드를 통한 본인인증 및 티켓출력	- 31 -
Figure 11 관리자 채널 1 - 패스워드를 통한 관리자 로그인	- 32 -
Figure 12 관리자 채널 2 - 관리자 메뉴와 저장된 DB출력	- 33 -
Figure 13 관리자 채널 3 - 관리자 메뉴와 DB저장	- 33 -
Figure 14 지정된 루트에 날짜별로 저장된 DB	- 34 -
Figure 15 2021.12.22일자 예약내역이 담긴 csv파일	- 34 -