

[4-2. 현금흐름 계산2]

1. 필요한 모듈 import

- 기존 작성한 모듈 및 필요 파일 확인
 - FCHotel_FSmodeling_assumption.xlsx
 - m00_general_function.py
 - m01_assumption.py
 - m02_index.py
 - m03_funding.py
 - m04_operating_income.py
 - m05_operating_cost.py
 - m06_facility_cost.py

```
In [ ]: import pandas as pd
pd.set_option('display.max_rows', 30)
pd.set_option('display.max_columns', 100)
pd.set_option('display.max_colwidth', 20)
pd.set_option('display.width', 300)

# DataFrame의 출력을 확장하여 한 줄로 계속 출력되도록 설정
pd.set_option('display.expand_frame_repr', True)

from m01_assumption import assumption
from m02_index import index
from m03_funding import funding
from m04_operating_income import operating_income
from m05_operating_cost import operating_cost
from m06_facility_cost import facility_cost
```

2. 운영수입 현금흐름 작성

2-1. operating_income 객체 확인

```
In [ ]: operating_income.keys()
```

```
Out[ ]: dict_keys(['TypeA', 'TypeB', 'TypeC', 'Total'])
```

```
In [ ]: operating_income['TypeA']
```

Out []:

	총객실수	사용불가객실수	판매가능객실수	객실판매비율	판매객실수	판매단가	객실수입
2023-12-31	0	0	0	0.00	0	0	0
2024-01-31	1240	0	1240	0.90	1116	130000	145080000
2024-02-29	1160	0	1160	0.85	986	130000	128180000
2024-03-31	1240	0	1240	0.80	992	100000	99200000
2024-04-30	1200	0	1200	0.80	960	100000	96000000
...
2026-09-30	1200	0	1200	0.80	960	100000	96000000
2026-10-31	1240	0	1240	0.80	992	100000	99200000
2026-11-30	1200	0	1200	0.70	840	100000	84000000
2026-12-31	1240	0	1240	0.90	1116	130000	145080000
2027-01-31	0	0	0	0.00	0	0	0

38 rows x 7 columns

2-2. operating_income 객체의 현금흐름 반영

In []:

```
#### cashflow, balance 객체 초기화 ####
cashflow = pd.DataFrame(
    columns = ['date', 'categoryA', 'categoryB', 'categoryC', '입금금액', '출금금액']
)

balance = pd.DataFrame({
    '기초현금': [0] * len(index['model']),
    '입금금액': [0] * len(index['model']),
    '출금금액': [0] * len(index['model']),
})
```

```

        '기말현금': [0] * len(index['model']),
    },
    index=index['model']
)
room_type_list = list(assumption['business_overview']['객실수'].keys())

#### cashflow, balance 작성 ####
idx = 0
cash_balance = 0
for dt in index['model']:
    #### 0. 기초현금 계산
    balance.loc[dt, '기초현금'] = cash_balance

    #### 1. 자금조달소요
    ## 1-1. 자기자본 유입
    amount = funding['자기자본'].loc[dt, '자기자본유입']
    if amount > 0:
        cashflow.loc[idx] = [dt, '자금조달', '자기자본', '자기자본유입', amount, 0]
        balance.loc[dt, '입금금액'] += amount
        idx += 1

    ## 1-2. 차입금 유입
    amount = funding['차입금'].loc[dt, '차입금유입']
    if amount > 0:
        cashflow.loc[idx] = [dt, '자금조달', '차입금', '차입금유입', amount, 0]
        balance.loc[dt, '입금금액'] += amount
        idx += 1

    ## 1-3. 자산매입
    amount = funding['자산매입'].loc[dt, '자산매입']
    if amount > 0:
        cashflow.loc[idx] = [dt, '자산매입', '자산매입', '매입대금지출', 0, amount]
        balance.loc[dt, '출금금액'] += amount
        idx += 1

```

```

amount = funding['자산매입'].loc[dt, '매입부수비용']
if amount > 0:
    cashflow.loc[idx] = [dt, '자산매입', '매입부수비용', '부수비용지출', 0, amount]
    balance.loc[dt, '출금금액'] += amount
    idx += 1

#### 2. 운영수입
amount = operating_income['TypeA'].loc[dt, '객실수입']
if amount > 0:
    cashflow.loc[idx] = [dt, '운영수입', '객실수입', 'TypeA', amount, 0]
    balance.loc[dt, '입금금액'] += amount
    idx += 1

amount = operating_income['TypeB'].loc[dt, '객실수입']
if amount > 0:
    cashflow.loc[idx] = [dt, '운영수입', '객실수입', 'TypeB', amount, 0]
    balance.loc[dt, '입금금액'] += amount
    idx += 1

amount = operating_income['TypeC'].loc[dt, '객실수입']
if amount > 0:
    cashflow.loc[idx] = [dt, '운영수입', '객실수입', 'TypeC', amount, 0]
    balance.loc[dt, '입금금액'] += amount
    idx += 1

#### 9. 기말현금 계산
cash_balance = (
    balance.loc[dt, '기초현금'] + balance.loc[dt, '입금금액'] - balance.loc[dt, '출금금액']
)
balance.loc[dt, '기말현금'] = cash_balance

```

2-3. 현금흐름 반영 결과 확인

```
In [ ]: cashflow
```

Out []:

	date	categoryA	categoryB	categoryC	입금금액	출금금액
0	2023-12-31	자금조달	자기자본	자기자본유입	100000000000	0
1	2023-12-31	자금조달	차입금	차입금유입	100000000000	0
2	2023-12-31	자산매입	자산매입	매입대금지출	0	180000000000
3	2023-12-31	자산매입	매입부수비용	부수비용지출	0	100000000000
4	2024-01-31	운영수입	객실수입	TypeA	1450800000	0
...
104	2026-11-30	운영수입	객실수입	TypeB	1008000000	0
105	2026-11-30	운영수입	객실수입	TypeC	840000000	0
106	2026-12-31	운영수입	객실수입	TypeA	1450800000	0
107	2026-12-31	운영수입	객실수입	TypeB	1674000000	0
108	2026-12-31	운영수입	객실수입	TypeC	1395000000	0

109 rows × 6 columns

```
In [ ]: cashflow[cashflow['categoryC'] == 'TypeA']
```

Out []:

	date	categoryA	categoryB	categoryC	입금금액	출금금액
4	2024-01-31	운영수입	객실수입	TypeA	145080000	0
7	2024-02-29	운영수입	객실수입	TypeA	128180000	0
10	2024-03-31	운영수입	객실수입	TypeA	99200000	0
13	2024-04-30	운영수입	객실수입	TypeA	96000000	0
16	2024-05-31	운영수입	객실수입	TypeA	99200000	0
...
94	2026-08-31	운영수입	객실수입	TypeA	145080000	0
97	2026-09-30	운영수입	객실수입	TypeA	96000000	0
100	2026-10-31	운영수입	객실수입	TypeA	99200000	0
103	2026-11-30	운영수입	객실수입	TypeA	84000000	0
106	2026-12-31	운영수입	객실수입	TypeA	145080000	0

35 rows x 6 columns

In []:

balance

Out []:

	기초현금	입금금액	출금금액	기말현금
2023-12-31	0	200000000000	190000000000	10000000000
2024-01-31	10000000000	444230000	0	1444230000
2024-02-29	1444230000	392080000	0	1836310000
2024-03-31	1836310000	305040000	0	2141350000
2024-04-30	2141350000	295200000	0	2436550000
...
2026-09-30	12047150000	301200000	0	12348350000
2026-10-31	12348350000	311240000	0	12659590000
2026-11-30	12659590000	268800000	0	12928390000
2026-12-31	12928390000	451980000	0	13380370000
2027-01-31	13380370000	0	0	13380370000

38 rows x 4 columns

3. 운영수입 계산 코드 축약하기

In []:

```
##### cashflow, balance 객체 초기화 #####
cashflow = pd.DataFrame(
    columns = ['date', 'categoryA', 'categoryB', 'categoryC', '입금금액', '출금금액']
)

balance = pd.DataFrame({
    '기초현금': [0] * len(index['model']),
    '입금금액': [0] * len(index['model']),
    '출금금액': [0] * len(index['model']),
```



```

        '기말현금': [0] * len(index['model']),
    },
    index=index['model']
)
room_type_list = list(assumption['business_overview']['객실수'].keys())

#### cashflow, balance 작성 ####
idx = 0
cash_balance = 0
for dt in index['model']:
    #### 0. 기초현금 계산
    balance.loc[dt, '기초현금'] = cash_balance

    #### 1. 자금조달소요
    ## 1-1. 자기자본 유입
    amount = funding['자기자본'].loc[dt, '자기자본유입']
    if amount > 0:
        cashflow.loc[idx] = [dt, '자금조달', '자기자본', '자기자본유입', amount, 0]
        balance.loc[dt, '입금금액'] += amount
        idx += 1

    ## 1-2. 차입금 유입
    amount = funding['차입금'].loc[dt, '차입금유입']
    if amount > 0:
        cashflow.loc[idx] = [dt, '자금조달', '차입금', '차입금유입', amount, 0]
        balance.loc[dt, '입금금액'] += amount
        idx += 1

    ## 1-3. 자산매입
    amount = funding['자산매입'].loc[dt, '자산매입']
    if amount > 0:
        cashflow.loc[idx] = [dt, '자산매입', '자산매입', '매입대금지출', 0, amount]
        balance.loc[dt, '출금금액'] += amount
        idx += 1

```

```

amount = funding['자산매입'].loc[dt, '매입부수비용']
if amount > 0:
    cashflow.loc[idx] = [dt, '자산매입', '매입부수비용', '부수비용지출', 0, amount]
    balance.loc[dt, '출금금액'] += amount
    idx += 1

#### 2. 운영수입
for room_type in room_type_list:
    amount = operating_income[room_type].loc[dt, '객실수입']
    if amount > 0:
        cashflow.loc[idx] = [dt, '운영수입', '객실수입', room_type, amount, 0]
        balance.loc[dt, '입금금액'] += amount
        idx += 1

#### 9. 기말현금 계산
cash_balance = (
    balance.loc[dt, '기초현금'] + balance.loc[dt, '입금금액'] - balance.loc[dt, '출금금액']
)
balance.loc[dt, '기말현금'] = cash_balance

```

In []: cashflow

Out []:

	date	categoryA	categoryB	categoryC	입금금액	출금금액
0	2023-12-31	자금조달	자기자본	자기자본유입	100000000000	0
1	2023-12-31	자금조달	차입금	차입금유입	100000000000	0
2	2023-12-31	자산매입	자산매입	매입대금지출	0	180000000000
3	2023-12-31	자산매입	매입부수비용	부수비용지출	0	10000000000
4	2024-01-31	운영수입	객실수입	TypeA	145080000	0
...
104	2026-11-30	운영수입	객실수입	TypeB	100800000	0
105	2026-11-30	운영수입	객실수입	TypeC	84000000	0
106	2026-12-31	운영수입	객실수입	TypeA	145080000	0
107	2026-12-31	운영수입	객실수입	TypeB	167400000	0
108	2026-12-31	운영수입	객실수입	TypeC	139500000	0

109 rows x 6 columns

4. 구분심표 표기 함수 작성

4-1. 구분심표 표기 코드 작성하기

In []:

```
df_copy = balance.copy()
for key, item in balance.items():
    df_copy[key] = item.apply(lambda x: f"{x:,.0f}")
df_copy
```

Out []:

	기초현금	입금금액	출금금액	기말현금
2023-12-31	0	20,000,000,000	19,000,000,000	1,000,000,000
2024-01-31	1,000,000,000	444,230,000	0	1,444,230,000
2024-02-29	1,444,230,000	392,080,000	0	1,836,310,000
2024-03-31	1,836,310,000	305,040,000	0	2,141,350,000
2024-04-30	2,141,350,000	295,200,000	0	2,436,550,000
...
2026-09-30	12,047,150,000	301,200,000	0	12,348,350,000
2026-10-31	12,348,350,000	311,240,000	0	12,659,590,000
2026-11-30	12,659,590,000	268,800,000	0	12,928,390,000
2026-12-31	12,928,390,000	451,980,000	0	13,380,370,000
2027-01-31	13,380,370,000	0	0	13,380,370,000

38 rows x 4 columns

In []:

```
df_copy.loc['2024-02-29', '기초현금']
```

Out []: '1,444,230,000'

4-2. 구분실표 표기 함수 작성하기

```
In [ ]: from pandas.api.types import is_numeric_dtype

def df_seperator(df):
    df_copy = df.copy()
    for key, item in df.items():
        if is_numeric_dtype(item):
            df_copy[key] = item.fillna(0).apply(lambda x: f"{x:,.0f}")
    return df_copy
```

```
In [ ]: df_seperator(balance)
```

Out []:

	기초현금	입금금액	출금금액	기말현금
2023-12-31	0	20,000,000,000	19,000,000,000	1,000,000,000
2024-01-31	1,000,000,000	444,230,000	0	1,444,230,000
2024-02-29	1,444,230,000	392,080,000	0	1,836,310,000
2024-03-31	1,836,310,000	305,040,000	0	2,141,350,000
2024-04-30	2,141,350,000	295,200,000	0	2,436,550,000
...
2026-09-30	12,047,150,000	301,200,000	0	12,348,350,000
2026-10-31	12,348,350,000	311,240,000	0	12,659,590,000
2026-11-30	12,659,590,000	268,800,000	0	12,928,390,000
2026-12-31	12,928,390,000	451,980,000	0	13,380,370,000
2027-01-31	13,380,370,000	0	0	13,380,370,000

38 rows x 4 columns

```
In [ ]: df_seperator(cashflow)
```

Out []:

	date	categoryA	categoryB	categoryC	입금금액	출금금액
0	2023-12-31	자금조달	자기자본	자기자본유입	10,000,000,000	0
1	2023-12-31	자금조달	차입금	차입금유입	10,000,000,000	0
2	2023-12-31	자산매입	자산매입	매입대금지출	0	18,000,000,000
3	2023-12-31	자산매입	매입부수비용	부수비용지출	0	1,000,000,000
4	2024-01-31	운영수입	객실수입	TypeA	145,080,000	0
...
104	2026-11-30	운영수입	객실수입	TypeB	100,800,000	0
105	2026-11-30	운영수입	객실수입	TypeC	84,000,000	0
106	2026-12-31	운영수입	객실수입	TypeA	145,080,000	0
107	2026-12-31	운영수입	객실수입	TypeB	167,400,000	0
108	2026-12-31	운영수입	객실수입	TypeC	139,500,000	0

109 rows x 6 columns

4-3. general_function 파일에 함수 추가

- 위에서 작성한 "df_seperator" 함수를 "m00_general_function.py" 파일에 추가

In []: