[3-2. 운영수입 설정]

1. 필요한 모듈 import 및 기본 설정

1-1. 필요한 모듈 import

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
```

1-2. assumption 데이터 확인

In [ ]:
```
assumption['business_overview']['객실수']
```

Out[ ]:
```
{'TypeA': 40, 'TypeB': 40, 'TypeC': 20}
```

In [ ]:
```
pd.DataFrame(assumption['monthly_occ_rate'])
```

Out[ ]:

	TypeA	TypeB	TypeC
1	0.90	0.90	0.85
2	0.85	0.85	0.80
3	0.80	0.80	0.70
4	0.80	0.80	0.70
5	0.80	0.80	0.75
6	0.70	0.70	0.70
7	0.90	0.90	0.85
8	0.90	0.90	0.85
9	0.80	0.80	0.75
10	0.80	0.80	0.75
11	0.70	0.70	0.70
12	0.90	0.90	0.90

In [ ]:
```
pd.DataFrame(assumption['monthly_price'])
```

Out[ ]:

	TypeA	TypeB	TypeC
1	130000	150000	250000
2	130000	150000	250000
3	100000	120000	200000
4	100000	120000	200000
5	100000	120000	200000
6	100000	120000	200000
7	130000	150000	250000
8	130000	150000	250000
9	100000	120000	200000
10	100000	120000	200000
11	100000	120000	200000
12	130000	150000	250000

1-3. 빈 operating_income 딕셔너리 설정

In [ ]:
```
operating_income = {}
```

2. 객실별 운영수입 계산

2-1. TypeA room의 객실 운영수입 계산

In [ ]:
```python
from datetime import datetime

dt = datetime(2024, 1, 31)
room_type = 'TypeA'
dct = {}
```

In [ ]:
```
# 총객실수
dct['총객실수'] = (
    assumption['business_overview']['객실수'][room_type] *
    index['days'].loc[dt, '월간일수']
)

# 사용불가객실수(대수선 기간)
if dt in index['수선' + room_type]:
    dct['사용불가객실수'] = (
        assumption['business_overview']['객실수'][room_type] *
        index['days'].loc[dt, '월간일수']
    )
else:
    dct['사용불가객실수'] = 0

# 판매가능객실수
dct['판매가능객실수'] = dct['총객실수'] - dct['사용불가객실수']

# 객실판매비율(OCC rate)
dct['객실판매비율'] = assumption['monthly_occ_rate'][room_type][dt.month]

# 판매객실수
dct['판매객실수'] = round(dct['판매가능객실수'] * dct['객실판매비율'])

# 판매단가
dct['판매단가'] = assumption['monthly_price'][room_type][dt.month]

# 객실수입
dct['객실수입'] = dct['판매객실수'] * dct['판매단가']
```

In [ ]:
```
dct
```

Out[ ]:
```
{'총객실수': 1240,
 '사용불가객실수': 0,
 '판매가능객실수': 1240,
 '객실판매비율': 0.9,
 '판매객실수': 1116,
 '판매단가': 130000,
 '객실수입': 145080000}
```

2-2. for문을 활용하여 운영기간별, room type별 데이터 계산

In [ ]:
```python
room_type_list = list(assumption['business_overview']['객실수'].keys())
for room_type in room_type_list:
    data = []
    for dt in index['model']:
        dct = {}

        if dt in index['operating']:
            # 총객실수
            dct['총객실수'] = (
                assumption['business_overview']['객실수'][room_type] *
                index['days'].loc[dt, '월간일수']
            )

            # 사용불가객실수(대수선 기간)
            if dt in index['수선' + room_type]:
                dct['사용불가객실수'] = (
                    assumption['business_overview']['객실수'][room_type] *
                    index['days'].loc[dt, '월간일수']
                )
            else:
                dct['사용불가객실수'] = 0

            # 판매가능객실수
            dct['판매가능객실수'] = dct['총객실수'] - dct['사용불가객실수']
```

```
            # 객실판매비율(OCC rate)
            dct['객실판매비율'] = assumption['monthly_occ_rate'][room_type][dt.month]

            # 판매객실수
            dct['판매객실수'] = round(dct['판매가능객실수'] * dct['객실판매비율'])

            # 판매단가
            dct['판매단가'] = assumption['monthly_price'][room_type][dt.month]

            # 객실수입
            dct['객실수입'] = dct['판매객실수'] * dct['판매단가']
        else:
            dct['총객실수'] = 0
            dct['사용불가객실수'] = 0
            dct['판매가능객실수'] = 0
            dct['객실판매비율'] = 0.0
            dct['판매객실수'] = 0
            dct['판매단가'] = 0
            dct['객실수입'] = 0

        data.append(dct)
    operating_income[room_type] = pd.DataFrame(data, index=index['model'])
```

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

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

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 × 7 columns

3. 구분값 별 Total 수치 계산

In [ ]:
```
data = {}
data['총객실수'] = (
    operating_income['TypeA']['총객실수']
    + operating_income['TypeB']['총객실수']
    + operating_income['TypeC']['총객실수']
)
data['사용불가객실수'] = (
    operating_income['TypeA']['사용불가객실수']
    + operating_income['TypeB']['사용불가객실수']
```

```
    + operating_income['TypeC']['사용불가객실수']
)
data['판매가능객실수'] = (
    operating_income['TypeA']['판매가능객실수']
    + operating_income['TypeB']['판매가능객실수']
    + operating_income['TypeC']['판매가능객실수']
)
data['판매객실수'] = (
    operating_income['TypeA']['판매객실수']
    + operating_income['TypeB']['판매객실수']
    + operating_income['TypeC']['판매객실수']
)
data['객실수입'] = (
    operating_income['TypeA']['객실수입']
    + operating_income['TypeB']['객실수입']
    + operating_income['TypeC']['객실수입']
)
data['객실판매비율'] = pd.Series(
    [(data['판매객실수'].loc[idx] / data['판매가능객실수'].loc[idx]
        if data['판매가능객실수'].loc[idx] != 0
        else 0
        ) for idx in index['model']
    ],
    index = index['model']
)
data['판매단가'] = pd.Series(
    [(int(round(data['객실수입'].loc[idx] / data['판매객실수'].loc[idx]))
        if data['판매객실수'].loc[idx] != 0
        else 0
        ) for idx in index['model']
    ],
    index = index['model']
)

df_temp = pd.DataFrame(data)
```

```
df_temp = df_temp[['총객실수', '사용불가객실수', '판매가능객실수', '객실판매비율', '판매객실수', '판매단가', '객실수입']]

operating_income['Total'] = df_temp
```

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

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

In [ ]:
```
operating_income['Total']
```

Out[ ]:

	총객실수	사용불가객실수	판매가능객실수	객실판매비율	판매객실수	판매단가	객실수입
2023-12-31	0	0	0	0.00	0	0	0
2024-01-31	3100	0	3100	0.89	2759	161011	444230000
2024-02-29	2900	0	2900	0.84	2436	160952	392080000
2024-03-31	3100	0	3100	0.78	2418	126154	305040000
2024-04-30	3000	0	3000	0.78	2340	126154	295200000
...	...	...	...	...	...	...	...
2026-09-30	3000	0	3000	0.79	2370	127089	301200000
2026-10-31	3100	0	3100	0.79	2449	127089	311240000
2026-11-30	3000	0	3000	0.70	2100	128000	268800000
2026-12-31	3100	0	3100	0.90	2790	162000	451980000
2027-01-31	0	0	0	0.00	0	0	0

38 rows × 7 columns

4. operating_income.py 파일 작성

In [ ]:
```python
# m04_operating_income.py

import pandas as pd

from m01_assumption import assumption
from m02_index import index


operating_income = {}

#### 1. 객실별 운영수입 계산
room_type_list = list(assumption['business_overview']['객실수'].keys())
for room_type in room_type_list:
    data = []
    for dt in index['model']:
        dct = {}

        if dt in index['operating']:
            # 총객실수
            dct['총객실수'] = (
                assumption['business_overview']['객실수'][room_type] *
                index['days'].loc[dt, '월간일수']
            )

            # 사용불가객실수(대수선 기간)
            if dt in index['수선' + room_type]:
                dct['사용불가객실수'] = (
                    assumption['business_overview']['객실수'][room_type] *
                    index['days'].loc[dt, '월간일수']
                )
            else:
                dct['사용불가객실수'] = 0

            # 판매가능객실수
            dct['판매가능객실수'] = dct['총객실수'] - dct['사용불가객실수']
```

```
            # 객실판매비율(OCC rate)
            dct['객실판매비율'] = assumption['monthly_occ_rate'][room_type][dt.month]

            # 판매객실수
            dct['판매객실수'] = round(dct['판매가능객실수'] * dct['객실판매비율'])

            # 판매단가
            dct['판매단가'] = assumption['monthly_price'][room_type][dt.month]

            # 객실수입
            dct['객실수입'] = dct['판매객실수'] * dct['판매단가']
        else:
            dct['총객실수'] = 0
            dct['사용불가객실수'] = 0
            dct['판매가능객실수'] = 0
            dct['객실판매비율'] = 0.0
            dct['판매객실수'] = 0
            dct['판매단가'] = 0
            dct['객실수입'] = 0

        data.append(dct)
    operating_income[room_type] = pd.DataFrame(data, index=index['model'])


#### 2. 운영수입 합계 계산
data = {}
data['총객실수'] = (
    operating_income['TypeA']['총객실수']
    + operating_income['TypeB']['총객실수']
    + operating_income['TypeC']['총객실수']
)
data['사용불가객실수'] = (
    operating_income['TypeA']['사용불가객실수']
    + operating_income['TypeB']['사용불가객실수']
```

```
    + operating_income['TypeC']['사용불가객실수']
)
data['판매가능객실수'] = (
    operating_income['TypeA']['판매가능객실수']
    + operating_income['TypeB']['판매가능객실수']
    + operating_income['TypeC']['판매가능객실수']
)
data['판매객실수'] = (
    operating_income['TypeA']['판매객실수']
    + operating_income['TypeB']['판매객실수']
    + operating_income['TypeC']['판매객실수']
)
data['객실수입'] = (
    operating_income['TypeA']['객실수입']
    + operating_income['TypeB']['객실수입']
    + operating_income['TypeC']['객실수입']
)
data['객실판매비율'] = pd.Series(
    [(data['판매객실수'].loc[idx] / data['판매가능객실수'].loc[idx]
        if data['판매가능객실수'].loc[idx] != 0
        else 0
        ) for idx in index['model']
    ],
    index = index['model']
)
data['판매단가'] = pd.Series(
    [(int(round(data['객실수입'].loc[idx] / data['판매객실수'].loc[idx]))
        if data['판매객실수'].loc[idx] != 0
        else 0
        ) for idx in index['model']
    ],
    index = index['model']
)

df_temp = pd.DataFrame(data)
```

```
df_temp = df_temp[['총객실수', '사용불가객실수', '판매가능객실수', '객실판매비율', '판매객실수', '판매단가', '객실수입']]

operating_income['Total'] = df_temp
```