

[3-3. 데이터 조회 모듈 작성]

1. 기 작성된 코드 실행

```
In [ ]: import pandas as pd

pd.set_option('display.max_rows', 20)
pd.set_option('display.max_columns', 100)
pd.set_option('display.max_colwidth', 50)
pd.set_option('display.width', 300)
pd.set_option('display.expand_frame_repr', True)

from pandas import DataFrame, Series
from datetime import datetime, date
from openpyxl import load_workbook
```

```
In [ ]: # 컬럼명 위치 행번호
colname_row_number = 4

# 데이터 시작 행번호
start_row_number = 5

# 데이터 종료 구분값
last_row_value = "합 계"
```

```
In [ ]: filename = "분개장_샘플.xlsx"

wb = load_workbook(filename)
ws = wb.active
```

```
In [ ]: colnames = [x.value for x in ws[colname_row_number]]
print(colnames)

['일자', '전표번호', '계정코드', '계정과목', '적요', '차변', '대변', '구분', '거래처명']
```

```
In [ ]: datalist = []
        for row in ws.iter_rows(
            min_row=start_row_number,
            max_col=len(colnames),
            values_only=True
        ):
            if row[0] == last_row_value:
                break
            row = list(row)

            # "일자", "전표번호" 컬럼에 값이 있는 경우 해당 값을 일자, 전표번호 변수에 저장
            if row[0] is not None:
                일자 = row[0].split('/')
                일자 = date(2024, int(일자[0]), int(일자[1]))
                전표번호 = row[1]

            # row 리스트의 첫번째 값은 "일자" 변수에 저장된 값으로 대체
            row[0] = 일자

            # row 리스트의 두번째 값은 "전표번호" 변수에 저장된 값으로 대체
            row[1] = 전표번호

            # 차변 컬럼의 값이 None인 경우 해당 값을 "0"으로 대체
            if pd.isna(row[5]):
                row[5] = 0

            # 대변 컬럼의 값이 None인 경우 해당 값을 "0"으로 대체
            if pd.isna(row[6]):
                row[6] = 0

            # 전처리 완료된 리스트를 datalist에 추가
            datalist.append(row)

            # 전처리 완료된 데이터를 데이터프레임으로 전환
            journal = DataFrame(datalist, columns=colnames)
```

```
In [ ]: # 분류값 테이블 생성
        category_table = pd.DataFrame([
            ['자본금', '자본', '자본금', '자본금'],
            ['보통예금', '유동자산', '현금및현금성자산', '보통예금'],
            ['객실수입', '매출', '매출', '객실수입'],
            ['고객용품비', '판매비와관리비', '고객용품비', '고객용품비'],
            ['기타영업비용', '판매비와관리비', '판매비용', '기타영업비용'],
            ['판매수수료', '판매비와관리비', '판매비용', '판매수수료'],
            ['세금과공과', '판매비와관리비', '기타판관비', '세금과공과'],
            ['수도광열비', '판매비와관리비', '기타판관비', '수도광열비'],
            ['급여', '판매비와관리비', '인건비', '급여'],
            ['노무용역비', '판매비와관리비', '인건비', '노무용역비'],
            ['이자수입', '영업외수익', '영업외수익', '이자수입']
        ],
        columns=['구분', '대분류', '중분류', '소분류']
        )
        category_table.set_index('구분', inplace=True)
```

```
In [ ]: # journal 데이터프레임에 "대분류", "중분류", "소분류" 컬럼 생성
journal['대분류'] = None
journal['중분류'] = None
journal['소분류'] = None

for idx in journal.index:
    # journal 데이터프레임의 "계정과목" 컬럼 값 추출
    key = journal.loc[idx, '계정과목']

    # category_table에서 추출된 키값에 해당하는 "대분류", "중분류", "소분류"
    # 값을 journal에 대체
    journal.loc[idx, '대분류'] = category_table.loc[key, '대분류']
    journal.loc[idx, '중분류'] = category_table.loc[key, '중분류']
    journal.loc[idx, '소분류'] = category_table.loc[key, '소분류']
```

```
In [ ]: journal = journal[
    [
        '일자', '전표번호', '대분류', '중분류', '소분류', '계정코드',
        '계정과목', '적요', '차변', '대변', '구분', '거래처명'
    ]
]
```

2. 데이터 조회

```
In [ ]: # 매출 데이터
journal[journal['대분류'] == '매출']
```

Out []:

	일자	전표번호	대분류	중분류	소분류	계정코드	계정과목	적요	차변	대변	구분	거래처명
3	2024-01-03	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	330000000	대변	호텔스닷컴
9	2024-01-04	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스넷
11	2024-01-05	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	300000000	대변	호텔스닷컴
13	2024-01-06	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스넷
15	2024-01-07	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	310000000	대변	호텔스닷컴
...
531	2024-05-26	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	340000000	대변	호텔스넷
533	2024-05-27	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	390000000	대변	호텔스닷컴
535	2024-05-28	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	380000000	대변	호텔스넷
537	2024-05-29	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스닷컴
547	2024-05-30	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	290000000	대변	호텔스닷컴

149 rows x 12 columns

In []:

```
# 보통예금 데이터 합계
journal[journal['계정과목'] == '보통예금']
```

Out[]:

	일자	전표번호	대분류	중분류	소분류	계정코드	계정과목	적요	차변	대변	구분	거래처명
1	2024-01-02	00001	유동자산	현금및현금성자산	보통예금	10300	보통예금	자본입금	2000000000	0	대변	하나은행
2	2024-01-03	00001	유동자산	현금및현금성자산	보통예금	10300	보통예금	객실매출	33000000	0	차변	하나은행
5	2024-01-03	00002	유동자산	현금및현금성자산	보통예금	10300	보통예금	고객용품지출	0	20000000	대변	하나은행
7	2024-01-03	00003	유동자산	현금및현금성자산	보통예금	10300	보통예금	영업비용 A	0	15000000	대변	하나은행
8	2024-01-04	00001	유동자산	현금및현금성자산	보통예금	10300	보통예금	객실매출	25000000	0	차변	하나은행
...
539	2024-05-29	00002	유동자산	현금및현금성자산	보통예금	10300	보통예금	객실판매수수료	0	3600000	대변	하나은행
541	2024-05-29	00003	유동자산	현금및현금성자산	보통예금	10300	보통예금	객실판매수수료	0	3200000	대변	하나은행
543	2024-05-29	00004	유동자산	현금및현금성자산	보통예금	10300	보통예금	고객용품지출	0	16100000	대변	하나은행
545	2024-05-29	00005	유동자산	현금및현금성자산	보통예금	10300	보통예금	영업비용 E	0	15000000	대변	하나은행
546	2024-05-30	00001	유동자산	현금및현금성자산	보통예금	10300	보통예금	객실매출	29000000	0	차변	하나은행

274 rows x 12 columns

```
In [ ]: # 날짜 기준과 결합하여 조회
보통예금 = journal[journal['계정과목'] == '보통예금']
보통예금[
    (보통예금['일자'] >= date(2024, 2, 1)) &
    (보통예금['일자'] <= date(2024, 2, 29))
]
```

Out[]:

	일자	전표번호	대분류	중분류	소분류	계정코드	계정과목	적요	차변	대변	구분	거래처명
108	2024-02-01	00001	유동자산	현금및 현금성 자산	보통예금	10300	보통예금	객실 매출	330000000	0	차변	하나은행
110	2024-02-02	00001	유동자산	현금및 현금성 자산	보통예금	10300	보통예금	정기 예금 이자	4246575	0	차변	신한은행
112	2024-02-02	00002	유동자산	현금및 현금성 자산	보통예금	10300	보통예금	객실 매출	250000000	0	차변	하나은행
114	2024-02-03	00001	유동자산	현금및 현금성 자산	보통예금	10300	보통예금	객실 매출	300000000	0	차변	하나은행
117	2024-02-03	00002	유동자산	현금및 현금성 자산	보통예금	10300	보통예금	고객 용품 비지출	0	200000000	대변	하나은행
...
206	2024-02-29	00001	유동자산	현금및 현금성 자산	보통예금	10300	보통예금	객실 매출	320000000	0	차변	하나은행
209	2024-02-29	00002	유동자산	현금및 현금성 자산	보통예금	10300	보통예금	객실 판매 수수료	0	28500000	대변	하나은행
211	2024-02-29	00003	유동자산	현금및 현금성 자산	보통예금	10300	보통예금	객실 판매 수수료	0	44500000	대변	하나은행
213	2024-02-29	00004	유동자산	현금및 현금성 자산	보통예금	10300	보통예금	고객 용품 비지출	0	150000000	대변	하나은행
215	2024-02-29	00005	유동자산	현금및 현금성 자산	보통예금	10300	보통예금	영업 비용E	0	200000000	대변	하나은행

54 rows × 12 columns

3. 데이터 조회 모듈 작성

3-1. 단일 구분값 모듈 작성

```
In [ ]: class Journal:
    def __init__(self, journal_df):
        # 외부에서 입력받은 journal_df를 df 속성에 저장.
        self.df = journal_df

        # 아래에 작성된 계정과목 클래스를 초기화하여 "Journal" 클래스의
        # 인스턴스("self")를 인수로 전달함.
        # 이렇게 함으로써 "계정과목" 클래스는 "Journal" 클래스의 인스턴스를
        # 참조할 수 있음.
        self.계정과목 = self.계정과목(self)

    # "계정과목" 클래스는 "Journal" 클래스의 내부에 정의된 중첩 클래스
    class 계정과목:
        def __init__(self, parent_instance):
            self.parent_instance = parent_instance
            self.attr_name = "계정과목"

        # "__call__" 메서드는 "계정과목" 클래스의 인스턴스를 함수처럼 호출할 수 있게
        # 해줌.
        def __call__(self, lookup_value):
            df = self.parent_instance.df
            # "df" 변수는 "parent_instance" 즉 "Journal" 클래스의 "df" 속성을
            # 참조함.

            filtered_df = df[df[self.attr_name] == lookup_value]
            # "df"에서 "self.attr_name" 컬럼의 값을 "lookup_value"로 필터링

            return Journal(filtered_df)
            # 필터링 된 데이터프레임을 사용하여 새로운 "Journal" 인스턴스를 생성하고
            # 반환함.
```

```
In [ ]: # 사용 예시
journal_instance = Journal(journal)

filtered_journal = journal_instance.계정과목('판매수수료')

print(filtered_journal.df)
```


	일자	전표번호	대분류	중분류	소분류	계정코드	계정과목	
적요	차변	대변	구분	거래처명				
16	2024-01-07	00002	판매비와관리비	판매비용	판매수수료	80300	판매수수료	객실
판매수수료	4700000	0	차변	호텔스닷컴				
20	2024-01-08	00002	판매비와관리비	판매비용	판매수수료	80300	판매수수료	객실
판매수수료	3750000	0	차변	호텔스넷				
38	2024-01-13	00002	판매비와관리비	판매비용	판매수수료	80300	판매수수료	객실
판매수수료	4550000	0	차변	호텔스닷컴				
42	2024-01-14	00002	판매비와관리비	판매비용	판매수수료	80300	판매수수료	객실
판매수수료	4600000	0	차변	호텔스넷				
58	2024-01-19	00002	판매비와관리비	판매비용	판매수수료	80300	판매수수료	객실
판매수수료	3700000	0	차변	호텔스닷컴				
..	
...					
502	2024-05-19	00003	판매비와관리비	판매비용	판매수수료	80300	판매수수료	객실
판매수수료	5650000	0	차변	호텔스닷컴				
526	2024-05-25	00002	판매비와관리비	판매비용	판매수수료	80300	판매수수료	객실
판매수수료	4900000	0	차변	호텔스넷				
528	2024-05-25	00003	판매비와관리비	판매비용	판매수수료	80300	판매수수료	객실
판매수수료	4550000	0	차변	호텔스닷컴				
538	2024-05-29	00002	판매비와관리비	판매비용	판매수수료	80300	판매수수료	객실
판매수수료	3600000	0	차변	호텔스넷				
540	2024-05-29	00003	판매비와관리비	판매비용	판매수수료	80300	판매수수료	객실
판매수수료	3200000	0	차변	호텔스닷컴				

[50 rows x 12 columns]

3-2. 구분값 별 모듈 확장

```
In [ ]: class Journal:
    def __init__(self, journal_df):
        self.df = journal_df
        self.전표번호 = self.전표번호(self)
        self.대분류 = self.대분류(self)
        self.중분류 = self.중분류(self)
        self.소분류 = self.소분류(self)
        self.계정코드 = self.계정코드(self)
        self.계정과목 = self.계정과목(self)
        self.거래처명 = self.거래처명(self)

    class 전표번호:
        def __init__(self, parent_instance):
            self.parent_instance = parent_instance
            self.attr_name = "전표번호"

        def __call__(self, lookup_value):
            df = self.parent_instance.df
            filtered_df = df[df[self.attr_name] == lookup_value]
            return Journal(filtered_df)

    class 대분류:
        def __init__(self, parent_instance):
            self.parent_instance = parent_instance
            self.attr_name = "대분류"

        def __call__(self, lookup_value):
```

```

        df = self.parent_instance.df
        filtered_df = df[df[self.attr_name] == lookup_value]
        return Journal(filtered_df)

class 중분류:
    def __init__(self, parent_instance):
        self.parent_instance = parent_instance
        self.attr_name = "중분류"

    def __call__(self, lookup_value):
        df = self.parent_instance.df
        filtered_df = df[df[self.attr_name] == lookup_value]
        return Journal(filtered_df)

class 소분류:
    def __init__(self, parent_instance):
        self.parent_instance = parent_instance
        self.attr_name = "소분류"

    def __call__(self, lookup_value):
        df = self.parent_instance.df
        filtered_df = df[df[self.attr_name] == lookup_value]
        return Journal(filtered_df)

class 계정코드:
    def __init__(self, parent_instance):
        self.parent_instance = parent_instance
        self.attr_name = "계정코드"

    def __call__(self, lookup_value):
        df = self.parent_instance.df
        filtered_df = df[df[self.attr_name] == lookup_value]
        return Journal(filtered_df)

class 계정과목:
    def __init__(self, parent_instance):
        self.parent_instance = parent_instance
        self.attr_name = "계정과목"

    def __call__(self, lookup_value):
        df = self.parent_instance.df
        filtered_df = df[df[self.attr_name] == lookup_value]
        return Journal(filtered_df)

class 거래처명:
    def __init__(self, parent_instance):
        self.parent_instance = parent_instance
        self.attr_name = "거래처명"

    def __call__(self, lookup_value):
        df = self.parent_instance.df
        filtered_df = df[df[self.attr_name] == lookup_value]
        return Journal(filtered_df)

```

```

In [ ]: # 사용 예시
        jnl = Journal(journal)

```

```
In [ ]: jn1.대분류("매출").df
```

Out []:

	일자	전표번호	대분류	중분류	소분류	계정코드	계정과목	적요	차변	대변	구분	거래처명
3	2024-01-03	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	330000000	대변	호텔스닷컴
9	2024-01-04	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스넷
11	2024-01-05	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	300000000	대변	호텔스닷컴
13	2024-01-06	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스넷
15	2024-01-07	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	310000000	대변	호텔스닷컴
...
531	2024-05-26	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	340000000	대변	호텔스넷
533	2024-05-27	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	390000000	대변	호텔스닷컴
535	2024-05-28	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	380000000	대변	호텔스넷
537	2024-05-29	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스닷컴
547	2024-05-30	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	290000000	대변	호텔스닷컴

149 rows × 12 columns

```
In [ ]: jn1.대분류("매출").거래처명("호텔스닷컴").df
```

Out []:

	일자	전표번호	대분류	중분류	소분류	계정코드	계정과목	적요	차변	대변	구분	거래처명
3	2024-01-03	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	330000000	대변	호텔스닷컴
11	2024-01-05	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	300000000	대변	호텔스닷컴
15	2024-01-07	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	310000000	대변	호텔스닷컴
23	2024-01-09	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	310000000	대변	호텔스닷컴
33	2024-01-11	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스닷컴
...
515	2024-05-23	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	270000000	대변	호텔스닷컴
525	2024-05-25	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스닷컴
533	2024-05-27	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	390000000	대변	호텔스닷컴
537	2024-05-29	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스닷컴
547	2024-05-30	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	290000000	대변	호텔스닷컴

79 rows × 12 columns

In []: `jn1.대분류("매출").소분류("객실수입").거래처명("호텔스닷컴").전표번호("00001").df`

Out []:

	일자	전표번호	대분류	중분류	소분류	계정코드	계정과목	적요	차변	대변	구분	거래처명
3	2024-01-03	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	330000000	대변	호텔스닷컴
11	2024-01-05	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	300000000	대변	호텔스닷컴
15	2024-01-07	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	310000000	대변	호텔스닷컴
23	2024-01-09	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	310000000	대변	호텔스닷컴
33	2024-01-11	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스닷컴
...
515	2024-05-23	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	270000000	대변	호텔스닷컴
525	2024-05-25	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스닷컴
533	2024-05-27	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	390000000	대변	호텔스닷컴
537	2024-05-29	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	250000000	대변	호텔스닷컴
547	2024-05-30	00001	매출	매출	객실수입	40100	객실수입	객실매출	0	290000000	대변	호텔스닷컴

79 rows × 12 columns

In []: