

Овчинников Степан Сергеевич ИУ5-51Б

1. «Каталог файлов» и «Файл» связаны соотношением один-ко-многим. Выведите список всех связанных каталогов и файлов, отсортированный по каталогам, сортировка по файлам произвольная.

2. «Каталог файлов» и «Файл» связаны соотношением один-ко-многим. Выведите список отделов с суммарным количеством строк в файлах в каждом каталоге, отсортированный по суммарному количеству.

3. «Каталог файлов» и «Файл» связаны соотношением многие-ко-многим. Выведите список всех каталогов, у которых в названии присутствует слово «каталог», и список находящихся в них файлов.

```
# используется для сортировки
from operator import itemgetter
from self import self

class File:
    """Файл"""
    def __init__(self, id, nazvan, stroki, katalog_id):
        self.id = id
        self.nazvan = nazvan
        self.stroki = stroki
        self.katalog_id = katalog_id

class Katalog:
    """Каталог"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class FilesKatalogs:
    """
    'Файл каталога' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """
    def __init__(self, katalog_id, file_id):
        self.katalog_id = katalog_id
        self.file_id = file_id

# Каталоги
katalogs = [
    Katalog(1, 'Каталог1'),
    Katalog(2, 'Каталог2'),
    Katalog(3, 'Каталог3'),
]

# Файлы
files = [
    File(1, '1', 2500, 1),
    File(2, '2', 3500, 2),
    File(3, '3', 4500, 3),
]
```

```

        File(4, '4', 3500, 3),
        File(5, '5', 2500, 1),
    ]
files_katalogs = [
    FilesKatalogs(1, 1),
    FilesKatalogs(2, 2),
    FilesKatalogs(3, 3),
    FilesKatalogs(3, 4),
    FilesKatalogs(3, 5),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(e.nazvan, e.stroki, d.name)
                   for d in catalogs
                   for e in files
                   if e.katalog_id == d.id]
    # Соединение данных многие-ко-многим
    many_to_many_tDir = [(d.name, ed.katalog_id, ed.file_id)
                         for d in catalogs
                         for ed in files_katalogs
                         if d.id == ed.katalog_id]
    many_to_many = [(e.nazvan, e.stroki, katalog_name)
                    for katalog_name, katalog_id, file_id in many_to_many_tDir
                    for e in files if e.id == file_id]
    print('Задание A1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)
    print('\n Задание A2 ')
    res_12_unsorted = []
    for d in catalogs:
        # Список
        d_files = list(filter(lambda i: i[2] == d.name, one_to_many))
    # Если не пустой
    if len(d_files) > 0:
        # Строки каталогов файла
        d_stroki = [stroki for _, stroki, _ in d_files]
        # Сумма строк файлов каталога
        d_stroki_sum = sum(d_stroki)
        res_12_unsorted.append((d.name, d_stroki_sum))
        # Сортировка по суммарной зарплате
        res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
        print(res_12)
        print('\n Задание A3')
        res_13 = {}
    # Перебираем все каталоги
    for d in catalogs:
        if 'Каталог' in d.name:
            # Список файлов каталога
            d_files = list(filter(lambda i: i[2] == d.name, many_to_many))
            # Только название файла
            d_files_names = [x for x, _, _ in d_files]
            # Добавляем результат в словарь
            # ключ - отдел, значение - список названий
            res_13[d.name] = d_files_names
    print(res_13)
    if __name__ == '__main__':
        main()

```

```
"D:\Учеба\Лабы РИП\venv\Scripts\python.exe" "D:/Учеба/Лабы РИП/RK1/RK1.py"
Задание A1
[('1', 2500, 'Каталог1'), ('5', 2500, 'Каталог1'), ('2', 3500, 'Каталог2'), ('3', 4500, 'Каталог3'), ('4', 3500, 'Каталог3')]

Задание A2
[['Каталог3', 8000]]

Задание A3
{'Каталог1': ['1'], 'Каталог2': ['2'], 'Каталог3': ['3', '4', '5']}

Process finished with exit code 0
```