

Текст программы:

```
from operator import itemgetter

class File:
    """Файл"""

    def __init__(self, id, name, weight_mb, catalog_id):
        self.id = id
        self.name = name
        self.weight_mb = weight_mb
        self.catalog_id = catalog_id

class Catalog:
    """Каталог файлов"""

    def __init__(self, id, path):
        self.id = id
        self.path = path

class FileCatalog:
    """
    'Файл в каталоге файлов' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """

    def __init__(self, catalog_id, file_id):
        self.catalog_id = catalog_id
        self.file_id = file_id

# Каталоги файлов
catalogs = [
    Catalog(1, 'C:\\'),
    Catalog(2, 'C:\\Users\\'),
    Catalog(3, 'C:\\Users\\User\\'),

    Catalog(11, 'D:\\'),
    Catalog(22, 'D:\\Users\\'),
    Catalog(33, 'A:\\Users\\Aboba\\'),
]

# Файлы
files = [
    File(1, 'MyBirthday.mp4ов', 14.9, 1),
    File(2, 'ActionCam2022-4-5.mp4', 1699.84, 2),
    File(3, 'Course-Work.docx', 0.013, 3),
    File(4, 'Task.pdf', 0.013, 3),
    File(5, 'Program.zipов', 100, 3),
]

# Файлы в каталогах файлов
files_catalogs = [
    FileCatalog(1, 1),
    FileCatalog(2, 2),
    FileCatalog(3, 3),
    FileCatalog(3, 4),
    FileCatalog(3, 5),

    FileCatalog(11, 1),
```

```

FileCatalog(22, 2),
FileCatalog(33, 3),
FileCatalog(33, 4),
FileCatalog(33, 5),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(f.name, f.weight_mb, c.path)
                    for c in catalogs
                    for f in files
                    if f.catalog_id == c.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(c.path, fc.catalog_id, fc.file_id)
                           for c in catalogs
                           for fc in files_catalogs
                           if c.id == fc.catalog_id]

    many_to_many = [(f.name, f.weight_mb, catalog_path)
                     for catalog_path, catalog_id, file_id in
many_to_many_temp
                     for f in files
                     if f.id == file_id]

    print('Задание Д1')
    res_11 = list(filter(lambda i: i[0][-2:] == "ов", one_to_many))
    print(res_11)

    print('\nЗадание Д2')
    print(sorted([(c.path, round(sum([otm[1] for otm in one_to_many if otm[2]
== c.path]) / (lambda x: 1 if x == 0 else x)(len([otm[1] for otm in
one_to_many if otm[2] == c.path]))) for c in catalogs], key=itemgetter(1),
reverse=True))

    print('\nЗадание Д3')
    res_13 = {}
    # Перебираем все каталоги
    for c in catalogs:
        if c.path[0] == 'A':
            # Список файлов каталога
            cat_fil = list(filter(lambda i: i[2] == c.path, many_to_many))
            # Только название файлов
            cat_fil_names = [x for x, _, _ in cat_fil]
            # Добавляем результат в словарь
            # ключ - каталог, значение - список файлов
            res_13[c.path] = cat_fil_names
    print(res_13)

    if __name__ == '__main__':
        main()

```

Результат выполнения:

```
C:\Users\kroll\PycharmProjects\BKIT\rk\rk1\venv\Scripts\python.exe C:\Users\kroll\PycharmProjects\BKIT\rk\rk1\main.py
```

Задание Д1

```
[('MyBirthday.mp4ов', 14.9, 'C:\\'), ('Program.zipов', 100, 'C:\\Users\\User\\')]
```

Задание Д2

```
[['C:\\Users\\', 1700], ['C:\\Users\\User\\', 33], ['C:\\', 15], ['D:\\', 0], ['D:\\Users\\', 0], ['A:\\Users\\Aboba\\', 0]]
```

Задание Д3

```
{'A:\\Users\\Aboba\\': ['Course-Work.docx', 'Task.pdf', 'Program.zipов']}
```

```
Process finished with exit code 0
```