

Вариант запросов: Г

Вариант предметной области: 15 (Файл, Каталог файлов)

1. «Каталог файлов» и «Файл» связаны соотношением один-ко-многим. Выведите список всех каталогов, у которых название начинается с буквы «А», и список содержащихся в них файлов.
2. «Каталог файлов» и «Файл» связаны соотношением один-ко-многим. Выведите список каталогов с максимальным размером файла в каждом каталоге, отсортированный по максимальному размеру.
3. «Каталог файлов» и «Файл» связаны соотношением многие-ко-многим. Выведите список всех связанных файлов и каталогов, отсортированный по каталогам, сортировка по файлам произвольная.

Исходный код программы:

```
2 from operator import itemgetter
3 import re
4
5 class File:
6     """Файл"""
7     def __init__(self, id, name, size, dir_id):
8         self.id = id
9         self.name = name
10        self.size = size
11        self.dir_id = dir_id
12
13 class Dir:
14     """Каталог файлов"""
15     def __init__(self, id, name):
16         self.id = id
17         self.name = name
18
19 class FileDir:
20     """Файлы в каталогах"""
21     def __init__(self, dir_id, file_id):
22         self.dir_id = dir_id
23         self.file_id = file_id
24
25 # Каталоги
26 dirs = [
27     Dir(1, 'Архитектура'),
28     Dir(2, 'универ'),
29     Dir(3, 'за день до конца света'),
```

```

30     Dir(22, 'абиотическое разнообразие'),
31
32     Dir(11, 'Архив странных фото'),
33     Dir(33, 'избранное жить'),
34 ]
35
36 # Файлы
37 files = [
38     File(1, 'супрематические примитивы', 165, 1),
39     File(2, 'рк', 142, 2),
40     File(3, 'я не помню когда это было', 100, 3),
41     File(4, 'Мультиагентный вентилятор', 210, 3),
42     File(5, 'многомерный скотч', 175, 3),
43     File(5, 'клеточный пожиратель', 201, 22),
44     File(5, 'пластмассовый мир', 110, 22),
45 ]
46
47 files_dirs = [
48     FileDir(1,1),
49     FileDir(2,2),
50     FileDir(3,3),
51     FileDir(3,4),
52     FileDir(3,5),
53     FileDir(22,2),
54
55     FileDir(11,1),
56     FileDir(33,3),
57     FileDir(33,4),
58     FileDir(33,5),
59 ]
60
61 def main():
62     """Основная функция"""
63
64     # Соединение данных один-ко-многим
65     one_to_many = [(f.name, f.size, d.name)
66                     for d in dirs
67                     for f in files
68                     if f.dir_id==d.id]
69
70     # Соединение данных многие-ко-многим
71     many_to_many_temp = [(d.name, fd.dir_id, fd.file_id)
72                            for d in dirs
73                            for fd in files_dirs
74                            if d.id==fd.dir_id]

```

```

75
76     many_to_many = [(f.name, f.size, dir_name)
77         for dir_name, dir_id, file_id in
many_to_many_temp
78         for f in files if f.id==file_id]
79
80     print('Задание A1')
81     res_11 = {}
82     selected_dirs = [one_dir[2] for one_dir in
one_to_many if one_dir[2].startswith('a') or
one_dir[2].startswith('A')]
83     for dir_name in selected_dirs:
84         files_in_dir = [(one_file[0],one_file[1]) for
one_file in one_to_many if one_file[2]==dir_name]
85         res_11.update({dir_name:files_in_dir})
86     print(res_11)
87
88     print('\nЗадание A2')
89     res_12_unsorted = []
90     for d in dirs:
91         d_files = list(filter(lambda i: i[2]==d.name,
one_to_many))
92         if len(d_files) > 0:
93             d_sizes = [size for _,size,_ in d_files]
94             d_size_max = max(d_sizes)
95             res_12_unsorted.append((d.name, d_size_max))
96
97     res_12 = sorted(res_12_unsorted, key=itemgetter(1),
reverse=True)
98     print(res_12)
99
100    print('\nЗадание A3')
101    res_13 = {}
102    dirs.sort(key=lambda one_dir: one_dir.name)
103
104    for d in dirs:
105        d_files = list(filter(lambda i: i[2]==d.name,
many_to_many))
106        d_files_names = [x for x,_,_ in d_files]
107        res_13[d.name] = d_files_names
108
109    print(res_13)
110
111    if __name__ == '__main__':
112        main()

```

Результат выполнения:

#### Задание A1

{'Архитектура': [( 'супрематические примитивы', 165)], 'абиотическое разнообразие': [( 'клеточный пожиратель', 201), ( 'пластмассовый мир', 110)]}

#### Задание A2

[( 'за день до конца света', 210), ( 'абиотическое разнообразие', 201), ( 'Архитектура', 165), ( 'универ', 142)]

#### Задание A3

{'Архив странных фото': [ 'супрематические примитивы'], 'Архитектура': [ 'супрематические примитивы'], 'абиотическое разнообразие': [ 'рк'], 'за день до конца света': [ 'я не помню когда это было', 'Мультиагентный вентилятор', 'многомерный скотч', 'клеточный пожиратель', 'пластмассовый мир'], 'избранное жить': [ 'я не помню когда это было', 'Мультиагентный вентилятор', 'многомерный скотч', 'клеточный пожиратель', 'пластмассовый мир'], 'универ': [ 'рк']}

#### Задание A1

```
{'Архитектура': [( 'супрематические примитивы', 165)], 'абиотическое разнообразие': [( 'клеточный пожиратель', 201), ( 'пластмассовый мир', 110)]}
```

#### Задание A2

```
[( 'за день до конца света', 210), ( 'абиотическое разнообразие', 201), ( 'Архитектура', 165), ( 'универ', 142)]
```

#### Задание A3

```
{'Архив странных фото': [ 'супрематические примитивы'], 'Архитектура': [ 'супрематические примитивы'], 'абиотическое разнообразие': [ 'рк'], 'за день до конца света': [ 'я не помню когда это было', 'Мультиагентный вентилятор', 'многомерный скотч', 'клеточный пожиратель', 'пластмассовый мир'], 'избранное жить': [ 'я не помню когда это было', 'Мультиагентный вентилятор', 'многомерный скотч', 'клеточный пожиратель', 'пластмассовый мир'], 'универ': [ 'рк']}
```