

**Вариант №25 «Раздел – Документ»****Вариант запросов Г.**

1. «Раздел» и «Документ» связаны соотношением один-ко-многим.

Выведите список всех разделов, у которых название начинается с буквы «l», и список хранящихся в них документов.

2. «Раздел» и «Документ» связаны соотношением один-ко-многим.

Выведите список разделов с максимальным размером документа в этом разделе, отсортированный по максимальному размеру.

3. «Раздел» и «Документ» связаны соотношением многие-ко-многим.

Выведите список всех связанных документов и разделов, отсортированный по разделам, сортировка по документам произвольная.

```
rk1 > 🐍 main.py > ⚙️ get_directories_with_documents
 1  class Directory:
 2      def __init__(self, id, name):
 3          self.id = id
 4          self.name = name
 5
 6
 7  class Document:
 8      def __init__(self, id, name, size, directory_id):
 9          self.id = id
10          self.name = name
11          self.size = size
12          self.directory_id = directory_id
13
14
15  class DirectoryDocument:
16      def __init__(self, directory_id, document_id):
17          self.directory_id = directory_id
18          self.document_id = document_id
19
20
21  directories = [
22      Directory(1, "university docx"),
23      Directory(2, "pcpl"),
24      Directory(3, "lab0"),
25      Directory(4, "lab1"),
26      Directory(5, "rk1")
27  ]
28
```

```
29
30     documents = [
31         Document(1, "lab0.py", 1, 3),
32         Document(2, "lab1.py", 3, 4),
33         Document(3, "lab100P.py", 4, 4),
34         Document(4, "rk1.py", 2, 5),
35         Document(5, "StudySchedule.docx", 350, 1)
36     ]
37
38
39     directories_documents = [
40         DirectoryDocument(1, 5),
41         DirectoryDocument(2, 1),
42         DirectoryDocument(2, 2),
43         DirectoryDocument(2, 3),
44         DirectoryDocument(2, 4),
45         DirectoryDocument(3, 1),
46         DirectoryDocument(4, 2),
47         DirectoryDocument(4, 3),
48         DirectoryDocument(5, 4)
49     ]
50
51
52     def get_directories_by_starting_letter(directories, documents, letter='1'):
53         result = {}
54
55         filtered_directories = [directory for directory in directories
56                                 if directory.name.lower().startswith(letter.lower())]
57
58         for directory in filtered_directories:
59             directory_documents = [doc for doc in documents if doc.directory_id == directory.id]
60             result[directory] = directory_documents
61
62     return result
63
64
65     def first_task():
66         print("Первое задание:\n")
67         result = get_directories_by_starting_letter(directories, documents, '1')
68
69         i = 1
70         for directory, docs in result.items():
71             print(f"{i}. {directory.name}:")
72             if docs:
73                 for doc in docs:
74                     print(f"\t- {doc.name}")
75             else:
76                 print("\tДокументов нет")
77             print()
78             i += 1
```

```

79
80
81 def filter_directories_by_max_size_of_documents(directories, documents):
82     directory_and_max_size = {}
83
84     for directory in directories:
85         directory_documents = [doc for doc in documents if doc.directory_id == directory.id]
86         if directory_documents:
87             max_document = max(directory_documents, key=lambda doc: doc.size)
88             directory_and_max_size[directory] = max_document.size
89         else:
90             directory_and_max_size[directory] = 0
91
92     result = sorted(directory_and_max_size.items(), key=lambda item: item[1], reverse=True)
93
94     return result
95
96
97 def second_task():
98     print("Второе задание:\n")
99     result = filter_directories_by_max_size_of_documents(directories, documents)
100
101    i = 1
102    for directory, max_size in result:
103        if max_size > 0:
104            print(f"{i}. {directory.name}: {max_size} KB")
105        else:
106            print(f"{i}. {directory.name}: нет документов")
107        i += 1
108
109    print()
110
111
112 def get_directories_with_documents(directories, documents, directories_documents):
113     result = {}
114
115     for directory in directories:
116         document_ids = [conn.document_id for conn in directories_documents
117                         if conn.directory_id == directory.id]
118
119         directory_documents = [doc for doc in documents if doc.id in document_ids]
120
121         result[directory] = directory_documents
122
123
124     return result

```

```

125
126 def third_task():
127     print("Третье задание:\n")
128     result = get_directories_with_documents(directories, documents, directories_documents)
129
130     i = 1
131     for directory, docs in result.items():
132         print(f"{i}. {directory.name}:")
133         if docs:
134             for doc in docs:
135                 print(f"\t- {doc.name}")
136         else:
137             print("\tДокументов нет")
138         print()
139         i += 1
140
141
142 def main():
143     first_task()
144     second_task()
145     third_task()
146
147
148 if __name__ == "__main__":
149     main()
150

```

## Результат выполнения

```
PS D:\Учёба\3 семестр\ПиКЯП\Shangin-PCPL-Labs-2025> py rk1/main.py
Первое задание:
```

1. lab0:
  - lab0.py
2. lab1:
  - lab1.py
  - lab100P.py

Второе задание:

1. university docx: 350 KB
2. lab1: 4 KB
3. rk1: 2 KB
4. lab0: 1 KB
5. pcpl: нет документов

Третье задание:

1. university docx:
  - StudySchedule.docx
2. pcpl:
  - lab0.py
  - lab1.py
  - lab100P.py
  - rk1.py
3. lab0:
  - lab0.py
4. lab1:
  - lab1.py
  - lab100P.py
5. rk1:
  - rk1.py

```
PS D:\Учёба\3 семестр\ПиКЯП\Shangin-PCPL-Labs-2025> █
```