

Условие:

Вариант Г, 25

25	Раздел	Документ
----	--------	----------

Вариант Г.

1. «Документ» и «Раздел» связаны соотношением один-ко-многим. Выведите список всех документов, у которых название начинается с буквы «А», и список разделов.
2. «Документ» и «Раздел» связаны соотношением один-ко-многим. Выведите список документов с максимальным количеством страниц одного раздела в каждом документе, отсортированный по максимальному количеству.
3. «Документ» и «Раздел» связаны соотношением многие-ко-многим. Выведите список всех связанных документов и разделов, отсортированный по документам, сортировка по разделам произвольная.

Текст кода:

```
1. #!/usr/bin/env python
2. from operator import itemgetter
3. class emps:
4.     """Разделы"""
5.     def __init__(self, id, name, stra, empa_id):
6.         self.id = id
7.         self.name = name
8.         self.stra = stra
9.         self.empa_id = empa_id
10. class Dep:
11.     """Документы"""
12.     def __init__(self, id, name):
13.         self.id = id
14.         self.name = name
15. class EmpDep:
16.     """'Разделы документов' для реализации связи многие-ко-многим"""
17.     def __init__(self, emps_id, empa_id):
18.         self.emps_id = emps_id
19.         self.empa_id = empa_id
20.
21. if __name__ == "__main__":
22.     Dep = [
23.         Dep(0, 'Отчет'),
24.         Dep(1, 'Курсовая'),
25.         Dep(2, 'Акт'),
26.         Dep(3, 'Автобиография'),
27.         Dep(4, 'Договор'),
```

```

28.     Dep(5, 'Реферат'),
29. ]
30.     emps = [
31.         emps(0, 'Начало', 2, 0),
32.         emps(1, 'Начало', 2, 1),
33.         emps(2, 'Введение', 2, 2),
34.         emps(3, 'Начало', 2, 3),
35.         emps(4, 'Начало', 2, 4),
36.         emps(5, 'Начало', 2, 5),
37.         emps(6, 'Констатирующая часть', 11, 2),
38.         emps(7, 'История', 9, 3),
39.         emps(8, 'Выводы', 13, 2),
40.         emps(9, 'Подписи', 18, 2),
41.         emps(10, 'Заключение', 50, 5),
42.         emps(11, 'Ведение', 1, 1),
43.     ]
44.     emps_empa = [
45.         EmpDep(1,1),
46.         EmpDep(2,2),
47.         EmpDep(3,3),
48.         EmpDep(3,4),
49.         EmpDep(3,5),
50.         EmpDep(11,1),
51.         EmpDep(22,2),
52.         EmpDep(33,3),
53.         EmpDep(33,4),
54.         EmpDep(33,5),
55.     ]
56.
57. # Соединение данных один-ко-многим
58.     one_to_many = [(d.name, d.stra, s.name)
59.                     for s in Dep
60.                     for d in emps
61.                     if d.empa_id==s.id]
62. # Соединение данных многие-ко-многим
63.     many_to_many_temp = [(s.name, ds.empa_id, ds.emps_id)
64.                            for s in Dep
65.                            for ds in emps_empa
66.                            if s.id==ds.empa_id]
67.
68.     many_to_many = [(d.name, d.stra, empa_name)
69.                      for empa_name, empa_id, emps_id in many_to_many_temp
70.                      for d in emps if d.id==emps_id]
71.     print("Задание Г1")
72.     res_11 = {}
73.     selected_empa = [one_book[2] for one_book in one_to_many if
74.                       one_book[2].startswith('a') or one_book[2].startswith('A')]
75.     for empa_name in selected_empa:
76.         emps_for_book = [(one_emps[0],one_emps[1]) for one_emps in
77.                           one_to_many if

```

```

78.     res_11.update({empa_name:emps_for_book})
79.     print(res_11)
80.     print()
81.
82.     print("Задание Г2")
83.     res_12_unsorted = []
84.     for s in Dep:
85.         s_emps = list(filter(lambda i: i[2]==s.name, one_to_many))
86.         if len(s_emps) > 0:
87.             s_stras = [stra for _,stra,_ in s_emps]
88.             s_stra_max = max(s_stras)
89.             res_12_unsorted.append((s.name, s_stra_max))
90.
91.     res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
92.     print(res_12)
93.     print()
94.
95.     print("Задание Г3")
96.     res_13 = {}
97.     Dep.sort(key=lambda one_empa: one_empa.name)
98.     for s in Dep:
99.         s_emps = list(filter(lambda i: i[2]==s.name, many_to_many))
100.        s_emps_names = [x for x,_,_ in s_emps]
101.        res_13[s.name] = s_emps_names
102.        print(res_13)

```

Результаты:

Задание Г1

{'Акт': [('Введение', 2), ('Констатирующая часть', 11), ('Выводы', 13), ('Подписи', 18)], 'Автобиография': [('Начало', 2), ('История', 9)]}

Задание Г2

[('Реферат', 50), ('Акт', 18), ('Автобиография', 9), ('Отчет', 2), ('Курсовая', 2), ('Договор', 2)]

Задание Г3

{'Автобиография': ['Начало'], 'Акт': ['Введение'], 'Договор': ['Начало'], 'Курсовая': ['Начало', 'Введение'], 'Отчет': [], 'Реферат': ['Начало']}