

## Рубежный контроль 1

### Вариант Г.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

33	Строка данных	Таблица данных
----	---------------	----------------

Код программы:

```
from operator import itemgetter
class Line:
    def __init__(self, id, name, val, dep_id):
        self.id = id
        self.name = name
        self.val = val
        self.dep_id = dep_id
class Table:
    def __init__(self, id, name):
        self.id = id
        self.name = name
class LineTable:
    def __init__(self, dep_id, line_id):
        self.dep_id = dep_id
        self.line_id = line_id

tables = [Table(1, 'Анализируемая таблица'),
          Table(2, 'В ожидании'),
          Table(3, 'Таблица пропущенных'),

          Table(11, 'Анализируемая таблица 2'),
          Table(22, 'В ожидании 2'),
          Table(33, 'Таблица пропущенных 2')]

lines = [Line(1, 'ПуМак', 604397, 1),
         Line(2, 'Шестерочка', 565819, 2),
         Line(3, 'Диски', 124781, 3),
         Line(4, 'Сложно - и точка', 617325, 4),
         Line(5, 'Ну-Ну', 981458, 5)]
```

```

lines_tables = [LineTable(1, 1),
                 LineTable(2, 2),
                 LineTable(3, 3),
                 LineTable(3, 4),
                 LineTable(3, 5),
                 LineTable(11, 1),
                 LineTable(22, 2),
                 LineTable(33, 3),
                 LineTable(33, 4),
                 LineTable(33, 5)]

def main():
    one_to_many = [(e.name, e.val, d.name)
                   for d in tables
                   for e in lines
                   if e.dep_id==d.id]

    many_to_many_temp = [(d.name, ed.dep_id, ed.line_id)
                          for d in tables
                          for ed in lines_tables
                          if d.id==ed.dep_id]

    many_to_many = [(l.name, l.val, dep_name)
                    for dep_name, dep_id, line_id in many_to_many_temp
                    for l in lines if l.id==line_id]

    print('Задание Г1')
    t_lines = list(filter(lambda i: i[2][0]=='A', one_to_many))
    print(t_lines)
    if len(t_lines) > 0:
        names = [name for name,_,_ in t_lines]
        print(names)

    print('\nЗадание Г2')
    ms = []
    for t in tables:
        t_lines = list(filter(lambda i: i[2]==t.name, one_to_many))
        mx = 0
        if len(t_lines) > 0:
            for line in t_lines:
                mx = max(mx, line[1])
            ms.append([mx, t_lines[0][2]])
    sms = sorted(ms, key=lambda x: x[0], reverse=True)
    print(sms)

    print('\nЗадание Г3')
    print(sorted(list((x[0], x[2]) for x in many_to_many),
                  key=itemgetter(1)))

if __name__ == '__main__':
    main()

```

Результат выполнения:

Задание Г1

```
[('РуМак', 604397, 'Анализируемая таблица')]  
['РуМак']
```

Задание Г2

```
[[604397, 'Анализируемая таблица'], [565819, 'В ожидании'], [124781,  
'Таблица пропущенных']]
```

Задание Г3

```
[('РуМак', 'Анализируемая таблица'), ('РуМак', 'Анализируемая таблица  
2'), ('Шестерочка', 'В ожидании'), ('Шестерочка', 'В ожидании 2'), ('Диски',  
'Таблица пропущенных'), ('Сложно - и точка', 'Таблица пропущенных'), ('Ну-  
Ну', 'Таблица пропущенных'), ('Диски', 'Таблица пропущенных 2'), ('Сложно - и  
точка', 'Таблица пропущенных 2'), ('Ну-Ну', 'Таблица пропущенных 2')]
```