## Рубежный контроль №1 Вариант

32 Колонка данных Таблица данных
----------------------------------

## 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.

## Текст программы

```
from operator import itemgetter
class DataColumn:
  def init (self, id, text data, size, table id):
    self.id = id
    self.text_data = text_data
    self.size = size
    self.table id = table id
class DataTable:
  def __init__(self, id, table_name):
    self.id = id
    self.table name = table name
class TableColumns:
  def __init__(self, column_id, table_id):
    self.column id = column id
    self.table id = table id
tables = [DataTable(1, "Clients"),
     DataTable(2, "Transactions"),
     DataTable(3, "Shipments"),
     DataTable(4, "Stocks"),
```

```
DataTable(5, "Losses"),
     DataTable(6, "Profits")]
columns = [DataColumn(11, "Name", 1024, 1),
      DataColumn(12, "Date", 128, 1),
      DataColumn(21, "Type", 1, 2),
      DataColumn(22, "Cost", 1024, 2),
      DataColumn(31, "Destination", 4096, 3),
      DataColumn(32, "Mass", 8, 3),
      DataColumn(41, "Company", 1024, 4),
      DataColumn(51, "Investment", 1024, 5),
      DataColumn(52, "Size", 1024, 5),
      DataColumn(61, "Investment", 1024, 6),
      DataColumn(62, "Size", 1024, 6)
      1
tabCol = [
  TableColumns(11, 1),
  TableColumns(12, 1),
  TableColumns(12, 3),
  TableColumns(12, 5),
  TableColumns(21, 2),
  TableColumns(22, 2),
  TableColumns(31, 3),
  TableColumns(32, 3),
  TableColumns(41, 1),
  TableColumns(41, 4),
  TableColumns(51, 5),
  TableColumns(52, 5),
  TableColumns(61, 6),
  TableColumns(62, 6)
]
def main():
  one_to_many = [(c.text_data, c.size, t.table_name)
  for c in columns
  for t in tables
  if c.table id == t.id]
  many to many temp = [(t.table name, tc.table id, tc.column id)
           for t in tables
           for tc in tabCol
           if t.id == tc.table id]
  many to many = [(c.text data, c.size, table name)
           for table name, table id, column id in many to many temp
           for c in columns if c.id == column id]
```

```
#F1
g1_temp = list(filter(lambda i: i[2][0] == 'S', one_to_many))
g1_t = list(set(list(x[2] for x in g1_temp)))
print(g1_t)
g1_c = list(x[0] for x in g1_temp)
print(g1_c)

#F2
g2_temp1 = list((x[2], x[1]) for x in one_to_many)
g2_temp2 = list(reversed(sorted(list(set(list(filter(lambda i: i[1] == max(list(x[1] for x in one_to_many if x[2] == i[0])), g2_temp1)))), key=itemgetter(1))))
print(g2_temp2)

#F3
print(sorted(list((x[0], x[2]) for x in many_to_many), key=itemgetter(1)))

if __name__ == '__main__':
    main()
```