**Код**

# вариант запроса Д  
# вариант предметной области 18 : музыкальное произведение - оркестр  
from operator import itemgetter  
  
  
class Composition:  
 # музыкальная композиция  
 def \_\_init\_\_(self, id, title, listeners, orchestra\_id):  
 self.id = id  
 self.title = title  
 self.listeners = listeners  
 self.orchestra\_id = orchestra\_id  
  
  
class Orchestra:  
 # оркестр  
 def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
  
class CompOrch:  
 # композиции оркестра для реализации связи  
 # многие-ко-многим  
 def \_\_init\_\_(self, orch\_id, comp\_id):  
 self.orch\_id = orch\_id  
 self.comp\_id = comp\_id  
  
# оркестры  
orchs = [  
 Orchestra(1, "Royal Concertgebouw Orchestra"),  
 Orchestra(2, "Berlin Philharmonic Orchestra"),  
 Orchestra(3, "Vienna Philharmonic Orchestra"),  
  
 Orchestra(4, "London Symphony Orchestra"),  
 Orchestra(5, "Chicago Symphony Orchestra"),  
 Orchestra(6, "Cleveland Orchestra")  
]  
  
# музыкальные композиции  
comps = [  
 Composition(1, "Sun Valley Serenade", 10000, 1),  
 Composition(2, "Toccata", 22000, 2),  
 Composition(3, "Love is blue", 45500, 2),  
 Composition(4, "Peer Gynt, Op. 23", 52000, 3),  
 Composition(5, "Eine kleine Nachtmusik, K.525: I. Allegro", 15000, 3),  
 Composition(6, "Symphony No.40 in G Minor, K.550: II. Andante", 100000, 3),  
 Composition(7, "Eine kleine Nachtmusik, K.525: IV. Rondo: Allegro", 10400, 3)  
]  
  
comps\_orchs = [  
 CompOrch(1, 1),  
 CompOrch(2, 2),  
 CompOrch(2, 3),  
 CompOrch(3, 4),  
 CompOrch(3, 5),  
 CompOrch(3, 6),  
 CompOrch(3, 7),  
  
 CompOrch(4, 1),  
 CompOrch(5, 2),  
 CompOrch(5, 3),  
 CompOrch(6, 4),  
 CompOrch(6, 5),  
 CompOrch(6, 6),  
 CompOrch(6, 7),  
]  
  
def main():  
 # соединение данных один-ко-многим  
 one\_to\_many = [(c.title, c.listeners, o.name)  
 for o in orchs  
 for c in comps  
 if c.orchestra\_id == o.id]  
  
 # соединение данных многие-ко-многим  
 many\_to\_many\_temp = [(o.name, co.orch\_id, co.comp\_id)  
 for o in orchs  
 for co in comps\_orchs  
 if o.id == co.orch\_id]  
  
 many\_to\_many = [(c.title, c.listeners, orch\_name)  
 for orch\_name, orch\_id, comp\_id in many\_to\_many\_temp  
 for c in comps if c.id == comp\_id]  
  
 print('Задание Д1')  
 res1 = []  
 for i in one\_to\_many:  
 if i[0][-2:] == "ro":  
 res1.append(i[0:3:2])  
 print(res1)  
  
 print('\nЗадание Д2')  
 res2\_unsorted = []  
 for o in orchs:  
 o\_comps = list(filter(lambda i: i[2] == o.name, one\_to\_many))  
 if len(o\_comps) > 0:  
 o\_listeners = [listeners for \_, listeners, \_ in o\_comps]  
 o\_listeners\_sum = sum(o\_listeners)  
 o\_listeners\_count = len(o\_listeners)  
 o\_listeners\_average = o\_listeners\_sum / o\_listeners\_count  
 res2\_unsorted.append((o.name, int(o\_listeners\_average)))  
 res2 = sorted(res2\_unsorted, key=itemgetter(1), reverse=True)  
 print(res2)  
  
 print('\nЗадание Д3')  
 res3 = {}  
 for o in orchs:  
 if o.name[0] == "C":  
 o\_comps = list(filter(lambda i: i[2] == o.name, many\_to\_many))  
 o\_comps\_titles = [x for x, \_, \_ in o\_comps]  
 res3[o.name] = o\_comps\_titles  
 print(res3)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

**Результат**

C:\Lab\_Python\Lab\_Python\RK1\venv\Scripts\python.exe C:/Lab\_Python/Lab\_Python/RK1/RK1.py

Задание Д1

[('Eine kleine Nachtmusik, K.525: I. Allegro', 'Vienna Philharmonic Orchestra'), ('Eine kleine Nachtmusik, K.525: IV. Rondo: Allegro', 'Vienna Philharmonic Orchestra')]

Задание Д2

[('Vienna Philharmonic Orchestra', 44350), ('Berlin Philharmonic Orchestra', 33750), ('Royal Concertgebouw Orchestra', 10000)]

Задание Д3

{'Chicago Symphony Orchestra': ['Toccata', 'Love is blue'], 'Cleveland Orchestra': ['Peer Gynt, Op. 23', 'Eine kleine Nachtmusik, K.525: I. Allegro', 'Symphony No.40 in G Minor, K.550: II. Andante', 'Eine kleine Nachtmusik, K.525: IV. Rondo: Allegro']}

Process finished with exit code 0

