



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ \_\_\_\_\_ РАДИОТЕХНИЧЕСКИЙ \_\_\_\_\_  
КАФЕДРА \_\_\_\_\_ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ \_\_\_\_\_

## ОТЧЕТ ПО РУБЕЖНОМУ КОНТРОЛЮ №1

Студент \_\_\_\_\_ Столярова Ольга Денисовна \_\_\_\_\_  
*фамилия, имя, отчество*

Группа \_\_\_\_\_ РТ5-51Б \_\_\_\_\_

Название предприятия \_\_\_\_\_ МГТУ им. Н. Э. Баумана \_\_\_\_\_

Студент \_\_\_\_\_ Столярова О.Д. \_\_\_\_\_

Преподаватель \_\_\_\_\_ Гапанюк Ю.Е. \_\_\_\_\_

2021 г.

## Цель работы

Работа с классами в Python, организация и реализация запросов.

## Задание

Рубежный контроль представляет собой разработку программы на языке Python, которая выполняет следующие действия:

1) Необходимо создать два класса данных, которые связаны отношениями один-ко-многим и многие-ко-многим.

Классы:

1. Класс «Музыкальное произведение», содержащий поля:

- ID произведения;
- Название;
- Длительность (количественный признак);
- ID записи об оркестре. (для реализации связи один-ко-многим)

Класс «Оркестр», содержащий поля:

- ID оркестра;
- Название оркестра.

(Для реализации связи многие-ко-многим) Класс «Музыкальные произведения оркестра», содержащий поля:

- ID записи о произведении;
- ID записи об оркестре.

2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.

3) Необходимо разработать следующие запросы:

1. «Оркестры» и «Музыкальные произведения» связаны соотношением один-ко-многим. Выведите список всех оркестров, у которых в названии присутствует слово «оркестр», и список их произведений.
2. «Оркестры» и «Музыкальные произведения» связаны соотношением один-ко-многим. Выведите список оркестров со средней длительностью произведений, отсортированный по средней длительности.
3. «Оркестры» и «Музыкальные произведения» связаны соотношением многие-ко-многим. Выведите список всех произведений, у которых название начинается с буквы «Т», и названия их оркестров.

При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

## Текст программы:

Класс «Оркестр»

orch.py

```
pack > orch.py > ...
1 class orch:
2     def __init__(self, id, name):
3         self.id = id
4         self.name = name
5
6
```

Класс «Музыкальные произведения»

comp.py

```
pack > comp.py > ...
1 class comp:
2     def __init__(self, id, mus, long, orch_id):
3         self.id = id
4         self.mus = mus
5         self.long = long
6         self.orch_id = orch_id
7
8
```

Класс «Музыкальные произведения оркестра»

comp\_orch.py

```
pack > comp_orch.py > ...
1 class comp_orch:
2     def __init__(self, comp_id, orch_id):
3         self.comp_id = comp_id
4         self.orch_id = orch_id
5
```

data.py

```

pack > data.py > ...
1  from pack import comp
2  from pack.comp import comp
3  from pack.orch import orch
4  from pack.comp_orch import comp_orch
5
6  comps = [
7      comp(1, 'Лебединая песня', 5.40, 1),
8      comp(2, 'Шербурские зонтики', 3.20, 2),
9      comp(3, 'Соната №14 "Лунная"', 7.15, 2),
10     comp(4, 'Жизнь', 6.28, 1),
11     comp(5, 'Тико-Тико', 2.49, 3),
12     comp(6, 'Not so far away', 5.28, 4),
13     comp(7, 'Ave Maria', 4.43, 3),
14     comp(8, 'Токката', 6.06, 1),
15     comp(9, 'Тебе одной', 3.56, 5)
16 ]
17
18 orchs = [
19     orch(1, "Большой симфонический оркестр"),
20     orch(2, "Лондонский оркестр"),
21     orch(3, "Персимфанс"),
22     orch(4, "Филармония"),
23     orch(5, "Симфонический оркестр Би-би-си")
24 ]
25
26 comps_orchs = [
27     comp_orch(1,1),
28     comp_orch(1,2),
29     comp_orch(2,2),
30     comp_orch(1,3),
31     comp_orch(3,3),
32     comp_orch(4,3),
33     comp_orch(7,4),
34     comp_orch(8,5),
35     comp_orch(1,5),
36     comp_orch(6,1),
37     comp_orch(7,2),
38     comp_orch(8,2),
39     comp_orch(5,3),
40     comp_orch(8,3),
41     comp_orch(9,3),
42     comp_orch(1,4),
43     comp_orch(7,5),
44     comp_orch(3,5)
45 ]

```

main.py

```

main.py > main
1 from pack import comp
2 from pack.comp import comp
3 from pack.orch import orch
4 from pack.comp_orch import comp_orch
5 from pack.data import *
6 from colorama import init
7 init()
8 from colorama import Fore, Back, Style
9
10 def main():
11
12     one_to_many = [(c.mus, c.long, o.name)
13                   for o in orchs
14                   for c in comps
15                   if c.orch_id==o.id]
16
17     many_to_many_temp = [(o.name, co.comp_id, co.orch_id)
18                          for o in orchs
19                          for co in comps_orchs
20                          if o.id==co.orch_id]
21
22     many_to_many = [(c.mus, c.long, orch_name)
23                    for orch_name, comp_id, orch_id in many_to_many_temp
24                    for c in comps if c.id==comp_id]
25
26     print(Back.CYAN + 'Задание A1 - список всех оркестров, в названии которых есть слово "оркестр"' + Style.RESET_ALL)
27
28     res11 = []
29
30     for i in one_to_many:
31         if i[2].find("оркестр") != -1:
32             int = [i[2], i[0]]
33             res11.append(int)
34
35     k = ""
36
37     for i in res11:
38         if k == "":
39             k = i[0]
40             print(i[0])
41             for i in res11:
42                 if k == i[0]:
43                     print(' ', i[1])
44             elif k != i[0]:
45                 k = i[0]
46
47         print(i[0])
48         for i in res11:
49             if k == i[0]:
50                 print(' ', i[1])
51
52     print(Fore.BLACK + Back.YELLOW + 'Задание A2 - средняя продолжительность произведений для каждого оркестра' + Style.RESET_ALL)
53
54     k = ""
55     s = 0
56     dl = 0
57
58     for i in one_to_many:
59         if k == "":
60             k = i[2]
61             for i in one_to_many:
62                 if k == i[2]:
63                     s += 1
64                     dl += i[1]
65                 print(k, "----", round(dl/s, 2))
66             elif k != i[2]:
67                 k = i[2]
68                 for i in one_to_many:
69                     if k == i[2]:
70                         s += 1
71                         dl += i[1]
72                     print(k, "----", round(dl/s, 2))
73
74     print(Back.MAGENTA + 'Задание A3 - список всех произведений, начинающихся на "Т"' + Style.RESET_ALL)
75
76     res13 = []
77     kkk = []
78
79     for i in many_to_many:
80         if i[0][0] == 'Т':
81             res13.append(i)
82
83     k = ""
84
85     for i in res13:
86         if k == "":
87             k = i[0]
88             kkk.append(i[0])
89             print(i[0])
90             for i in res13:

```

```

90         if k == i[0]:
91             print(' ', i[2])
92     elif k != i[0] and i[0] not in kkk:
93         k = i[0]
94         kkk.append(i[0])
95         print(i[0])
96     for i in res13:
97         if k == i[0]:
98             print(' ', i[2])
99
100
101
102
103
104 if __name__ == '__main__':
105     main()
106
107

```

## Экранные формы с примерами выполнения программы:

```

python.python-2021.10.1365161279\pythonFiles\lib\python\debugpy\launcher '49339' '--' 'c:\Users\olyas\python\PK1\main.py'
...
Задание A1 - список всех оркестров, в названии которых есть слово "оркестр"
Большой симфонический оркестр
    Лебединая песня
    Жизнь
    Токката
Лондонский оркестр
    Шербурские зонтики
    Соната №14 "Лунная"
Симфонический оркестр Би-би-си
    Тебе одной
Задание A2 - средняя продолжительность произведений для каждого оркестра
Большой симфонический оркестр ---- 5.91
Лондонский оркестр ---- 5.62
Персимфанс ---- 5.0
Филармония ---- 5.04
Симфонический оркестр Би-би-си ---- 4.87
Задание A3 - список всех произведений, начинающихся на "Т"
Токката
    Лондонский оркестр
    Персимфанс
    Симфонический оркестр Би-би-си
Тико-Тико
    Персимфанс
Тебе одной
    Персимфанс
PS C:\Users\olyas\python\PK1>

```

## Вывод

Продемонстрированы возможности работы с классами в Python, организации и реализации запросов.