

Студент: Нестеренко Иван

Группа: ИУ5-32Б

Рубежный контроль №2 по дисциплине «Базовые компоненты интернет-технологий»

Вариант №17Б

Текст программы:

Файл Conductors.py

```
class conductor():
    #дирижер
    def __init__(self, id_value, name_value, rating_value,
id_orchestra_value):
        self.id = id_value
        self.name = name_value
        self.rating = rating_value
        self.id_orchestra = id_orchestra_value
```

Файл Orchestras.py

```
class orchestra():
    #оркестра
    def __init__(self, id_value, number_value):
        self.id = id_value
        self.number = number_value
```

Файл CONORCH.py

```
class ConOrch():
    #дирижеры оркестра, для реализации связи многие-ко-
многим
    def __init__(self, id_con_val, id_orch_val):
        self.id_con = id_con_val
        self.id_orch = id_orch_val
```

Файл main.py

```
from Conductors import conductor
from Orchestras import orchestra
from CONORCH import ConOrch
```

```
from operator import itemgetter
```

```
#Вариант 17Б
```

```
#Оркестры
```

```
orchestras = [  
    orchestra(1, 'Оркестр №1'),  
    orchestra(2, 'Оркестр №2'),  
    orchestra(3, 'Оркестр №3'),  
    orchestra(11, 'не-Оркестр №1'),  
    orchestra(22, 'не-Оркестр №2'),  
    orchestra(33, 'не-Оркестр №3')  
]
```

```
#Дирижеры
```

```
conductors = [  
    conductor(1, 'Мурорский', 4.7, 3),  
    conductor(2, 'Глинка', 5.0, 3),  
    conductor(3, 'Верди', 3.9, 2),  
    conductor(4, 'Стравинский', 4.9, 2),  
    conductor(5, 'Крылов', 4.6, 2),  
    conductor(6, 'Гончаров', 4.1, 1)  
]
```

```
con_orch = [  
    ConOrch(1, 1),  
    ConOrch(2, 2),  
    ConOrch(3, 2),  
    ConOrch(4, 3),  
    ConOrch(5, 3),  
    ConOrch(6, 3),  
  
    ConOrch(1, 33),  
    ConOrch(2, 22),  
    ConOrch(3, 33),  
    ConOrch(4, 11),  
    ConOrch(5, 11),  
    ConOrch(6, 22)  
]
```

```
def task_1(one_to_many):
```

```
    try:
```

```
        result = sorted(one_to_many, key = itemgetter(0))  
        print("First task:")
```

```

        print(result)
    except(TypeError):
        raise TypeError("Enter the list[]")

def task_2(one_to_many):
    result = []
    for i in range(3):
        count_con = list(filter(lambda j: j[2] ==
orchestras[i].number, one_to_many))
        result.append((orchestras[i].number,
len(count_con)))
    result = sorted(result, key = itemgetter(1), reverse =
True)
    print("Second task:")
    print(result)
    return result

def task_3(many_to_many):
    result = []
    for i in many_to_many:
        if (i[0][-2:] == 'ОВ'):
            result.append(i)
    print("Third task:")
    print(result)
    return result

def main():

    #соединение данных один-ко-многим
    one_to_many = [(c.name, c.rating, orc.number)
        for orc in orchestras
        for c in conductors
        if c.id_orchestra == orc.id
    ]

    #соединение данных многие-ко-многим
    many_to_many_temp = [(orc.id, orc.number, corc.id_con)
        for orc in orchestras
        for corc in con_orch
        if orc.id == corc.id_orch
    ]

    many_to_many = [(c.name, c.rating, orch_num)
        for c in conductors

```

```

        for orch_id, orch_num, c_id in many_to_many_temp:
            if c_id == c_id:
                ]
            task_1(one_to_many)
            task_2(one_to_many)
            task_3(many_to_many)

if __name__ == "__main__":
    main()

```

Файл main.py

```

from unittest import TestCase, main
from main import task_1, task_2, task_3

class Testing_rk1(TestCase):
    def test_task_1(self):
        self.assertEqual(task_1([('Верди', 3.9, 'Оркестр
№2'),
                                ('Глинка', 5.0, 'Оркестр
№3'),
                                ('Крылов', 4.6, 'Оркестр
№2'),
                                ('Мурорский', 4.7, 'Оркестр
№3'),
                                ('Гончаров', 4.1, 'Оркестр
№1'),
                                ('Стравинский',
4.9, 'Оркестр №2')]),
                                [ ('Мурорский', 4.7, 'Оркестр №3'),
                                ('Глинка', 5.0, 'Оркестр
№3'),
                                ('Крылов', 4.6, 'Оркестр
№2'),
                                ('Верди', 3.9, 'Оркестр
№2'),
                                ('Гончаров', 4.1, 'Оркестр
№1'),
                                ('Стравинский',
4.9, 'Оркестр №2') ]))

    def test_task_1_error(self):

```

[illegible]

4.9, 'Оркестр №2'),
Оркестр №2')]),

№2'),
№2')])

('Стравинский',
('Стравинский', 4.9, 'не-
[('Глинка', 5.0, 'Оркестр №3'),
('Глинка', 5.0, 'не-Оркестр №3'),
('Крылов', 4.6, 'Оркестр №2'),
('Крылов', 4.6, 'не-Оркестр №2'),
('Стравинский', 4.9, 'Оркестр
('Стравинский', 4.9, 'не-Оркестр