

## Код программы.

```
import unittest
from operator import itemgetter
from main import conds, orcs, conds_orcs, main

class TestRK1Functions(unittest.TestCase):
    def test_task_E1(self):
        expected_result = [('Петров', 45000, 'Струнный оркестр'),
                           ('Ванин', 55000, 'Струнный оркестр'), ('Семенов', 45000, 'Струнный
оркестр')]
        one_to_many = [(c.fio, c.sal, o.name)
                        for o in orcs
                        for c in conds
                        if c.orc_id == o.id]
        filtered_result = list(filter(lambda i: i[2].find('Струнный') != -
1, one_to_many))
        self.assertEqual(filtered_result, expected_result)

    def test_task_E2(self):
        expected_result = [('Струнный оркестр', 48333.33), ('Джазовый
оркестр', 35000.0), ('Симфонический оркестр', 30000.0), ('Духовой
оркестр', 25000.0), ('Эстрадный оркестр', 25000.0), ('Военный оркестр',
25000.0)]
        one_to_many = [(c.fio, c.sal, o.name)
                        for o in orcs
                        for c in conds
                        if c.orc_id == o.id]
        res_12_unsorted = []
        for o in orcs:
            orc_conds = list(filter(lambda i: i[2] == o.name,
one_to_many))
            if len(orc_conds) > 0:
                o_sals = [sal for _, sal, _ in orc_conds]
                o_sals_sum = round(sum(o_sals) / len(orc_conds), 2)
                res_12_unsorted.append((o.name, o_sals_sum))

        res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
        self.assertEqual(res_12, expected_result)

    def test_task_E3(self):
        expected_result = [('Артамонов', 25000, 'Духовой оркестр'),
                           ('Абачков', 25000, 'Военный оркестр')]
        many_to_many_temp = [(o.name, co.orc_id, co.cond_id)
                              for o in orcs
```

```

        for co in conds_orcs
            if o.id == co.orc_id]
many_to_many = [(c.fio, c.sal, orc_name)
                 for orc_name, orc_id, cond_id in many_to_many_temp
                 for c in conds if c.id == cond_id]
filtered_result = list(filter(lambda i: i[0].find('A') != -1,
many_to_many))
self.assertEqual(filtered_result, expected_result)

if __name__ == '__main__':
    unittest.main()

```

## Результат выполнения

```

✓ Tests passed: 3 of 3 tests = 2ms
C:\Users\lerum\OneDrive\Desktop\pythonProject1\Scripts\python.exe "C:/Users/lerum/PyCharm/JetBrains/PyCharm Community Edition
Testing started at 17:22 ...

Ran 3 tests in 0.002s

OK
Launching unittests with arguments python -m unittest C:\Users\lerum\OneDrive\Desktop\paradigms\RK2\test_main.py in C:\Users\

<main.Orc object at 0x00000215533E5F10>

Process finished with exit code 0

```