Текст Программы

main.py

```
from operator import itemgetter


class File:

    def __init__(self, id, name, size, id_Cat):
        self.id = id
        self.name = name
        self.size = size
        self.id_Cat = id_Cat


class Catalog_Files:
    def __init__(self, id, name, size):
        self.id = id
        self.name = name
        self.size = size


class Fi_Cat:
    def __init__(self, id_Fi, id_Cat):
        self.id_Fi = id_Fi
        self.id_Cat = id_Cat


Files = [
    File(1, 'Laba_1', 2.4, 1),
    File(2, 'Laba_2', 2.1, 1),
    File(3, 'Laba_3', 1.8, 1),
    File(4, 'Курсовая', 3.5, 2),
    File(5, 'Макет', 4.0, 2),
    File(6, 'ДЗ_Физика', 1.4, 3)
]

Catalogs_Files = [
    Catalog_Files(1, 'Лабораторные', 5.0),
    Catalog_Files(2, 'ИУ-5', 7.5),
    Catalog_Files(3, 'Семестр_3', 1.4)

]

Fi_Cats = [
    Fi_Cat(1, 1),
    Fi_Cat(2, 1),
    Fi_Cat(3, 1),
    Fi_Cat(4, 2),
    Fi_Cat(5, 2),
    Fi_Cat(6, 3)
]


def Solution1(one_to_many):
    print('Задание А1')
    one_to_many_sort = sorted(one_to_many, key=itemgetter(0))
    return one_to_many_sort
```

```
def Solution2(one_to_many):
    print('Задание А2')
    one_to_many_unsorted = []
    for c in Catalogs_Files:
        # Список файлов каталога по
        c_files = list(filter(lambda i: i[2] == c.name, one_to_many))
        if len(c_files) > 0:
            size_files = [size for _, size, _ in c_files]
            # Суммируем обьемы файлов для каждого каталога
            sum_size = sum(size_files)
            one_to_many_unsorted.append((c.name, sum_size))
    one_to_many2_sort = sorted(one_to_many_unsorted, key=itemgetter(1),
reverse=True)
    return one_to_many2_sort


def Solution3(many_to_many):
    print('Задание А3')
    many_to_many_unsorted = {}
    for c in Catalogs_Files:
        if 'Лабо' in c.name:
            c_files = list(filter(lambda k: k[0] == c.name, many_to_many))
            c_files_names = [x for _, x, _ in c_files]
            many_to_many_unsorted[c.name] = c_files_names
    return many_to_many_unsorted


def main():
    one_to_many = [(f.name, f.size, c.name)
                   for f in Files
                   for c in Catalogs_Files
                   if f.id_Cat == c.id]
    print(one_to_many)
    many_to_many_temp = [(c.name, FC.id_Fi, FC.id_Cat)
                         for c in Catalogs_Files
                         for FC in Fi_Cats
                         if c.id == FC.id_Cat]
    many_to_many = [(Cat_name, f.name, f.size,)
                    for Cat_name, FC_id_Fi, FC_id_Cat in many_to_many_temp
                    for f in Files if f.id == FC_id_Fi]


    print(Solution1(one_to_many))
    print(Solution2(one_to_many))
    print(Solution3(many_to_many))
if __name__ == '__main__':
    main()
```

Tests.py

```
import unittest
from main import *


class TestRK2(unittest.TestCase):
    # Компьютеры
    files = [
        File(1, 'Laba_1', 2.4, 1),
        File(2, 'Laba_2', 2.1, 1),
        File(3, 'Laba_3', 1.8, 1),
        File(4, 'Курсовая', 3.5, 2),
```

```
        File(5, 'Макет', 4.0, 2),
        File(6, 'ДЗ_Физика', 1.4, 3)
    ]

    # Микропроцессоры
    catalog_files = [
        Catalog_Files(1, 'Лабораторные', 5.0),
        Catalog_Files(2, 'ИУ-5', 7.5),
        Catalog_Files(3, 'Семестр_3', 1.4)
    ]

    def test_A1(self):
        one_to_many = [(f.name, f.size, c.name)
                       for f in Files
                       for c in Catalogs_Files
                       if f.id_Cat == c.id]
        self.assertEqual(Solution1(one_to_many),
                         [('Laba_1', 2.4, 'Лабораторные'),
                          ('Laba_2', 2.1, 'Лабораторные'),
                          ('Laba_3', 1.8, 'Лабораторные'),
                          ('ДЗ_Физика', 1.4, 'Семестр_3'),
                          ('Курсовая', 3.5, 'ИУ-5'),
                          ('Макет', 4.0, 'ИУ-5')])

    def test_A2(self):
        one_to_many = [(f.name, f.size, c.name)
                       for f in Files
                       for c in Catalogs_Files
                       if f.id_Cat == c.id]
        self.assertEqual(Solution2(one_to_many),
                         [('ИУ-5', 7.5), ('Лабораторные', 6.3), ('Семестр_3',
1.4)])

    def test_A3(self):
        many_to_many_temp = [(c.name, FC.id_Fi, FC.id_Cat)
                             for c in Catalogs_Files
                             for FC in Fi_Cats
                             if c.id == FC.id_Cat]

        many_to_many = [(Cat_name, f.name, f.size,)
                        for Cat_name, FC_id_Fi, FC_id_Cat in
many_to_many_temp
                        for f in Files if f.id == FC_id_Fi]
        self.assertEqual(Solution3(many_to_many),
                         {'Лабораторные': ['Laba_1', 'Laba_2', 'Laba_3']})


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

Результаты

```
Ran 3 tests in 0.002s

OK
```