

# Текст программы из РК1 для модульного тестирования:

## Файл tasks.py

```
from operator import itemgetter

class File:
    def __init__(self, id, name, weight_mb, catalog_id):
        self.id = id
        self.name = name
        self.weight_mb = weight_mb
        self.catalog_id = catalog_id

class Catalog:
    def __init__(self, id, path):
        self.id = id
        self.path = path

class FileCatalog:
    def __init__(self, catalog_id, file_id):
        self.catalog_id = catalog_id
        self.file_id = file_id

catalogs = [
    Catalog(1, 'C:\\'),
    Catalog(2, 'C:\\Users\\'),
    Catalog(3, 'C:\\Users\\User\\'),

    Catalog(11, 'D:\\'),
    Catalog(22, 'D:\\Users\\'),
    Catalog(33, 'A:\\Users\\Aboba\\'),
]

files = [
    File(1, 'MyBirthday.mp4ов', 14.9, 1),
    File(2, 'ActionCam2022-4-5.mp4', 1699.84, 2),
    File(3, 'Course-Work.docx', 0.013, 3),
    File(4, 'Task.pdf', 0.013, 3),
    File(5, 'Program.zipов', 100, 3),
]

files_catalogs = [
    FileCatalog(1, 1),
    FileCatalog(2, 2),
    FileCatalog(3, 3),
    FileCatalog(3, 4),
    FileCatalog(3, 5),

    FileCatalog(11, 1),
    FileCatalog(22, 2),
    FileCatalog(33, 3),
    FileCatalog(33, 4),
    FileCatalog(33, 5),
]

one_to_many = [(f.name, f.weight_mb, c.path)
                for c in catalogs
                for f in files
                if f.catalog_id == c.id]

many_to_many_temp = [(c.path, fc.catalog_id, fc.file_id)
                      for c in catalogs
```

```

        for fc in files_catalogs
        if c.id == fc.catalog_id]

many_to_many = [(f.name, f.weight_mb, catalog_path)
                 for catalog_path, catalog_id, file_id in many_to_many_temp
                 for f in files
                 if f.id == file_id]

def task_1():
    res_11 = list(filter(lambda i: i[0][-2:] == "ов", one_to_many))
    return res_11

def task_2():
    return sorted([[c.path, round(
        sum([otm[1] for otm in one_to_many if otm[2] == c.path]) / (lambda x:
1 if x == 0 else x)(
        len([otm[1] for otm in one_to_many if otm[2] == c.path])))] for c
in catalogs], key=itemgetter(1),
                    reverse=True)

def task_3():
    res_13 = {}
    for c in catalogs:
        if c.path[0] == 'A':
            cat_fil = list(filter(lambda i: i[2] == c.path, many_to_many))
            cat_fil_names = [x for x, _, _ in cat_fil]
            res_13[c.path] = cat_fil_names
    return res_13

```

## Текст программы модульного тестирования:

### Файл tdd.py

```

import unittest
from tasks import task_1, task_2, task_3

task_1_result = [('MyBirthday.mp4ов', 14.9, 'C:\\'), ('Program.zipов', 100,
'C:\\Users\\User\\')]
task_2_result = [['C:\\Users\\', 1700], ['C:\\Users\\User\\', 33], ['C:\\',
15], ['D:\\', 0], ['D:\\Users\\', 0], ['A:\\Users\\Aboba\\', 0]]
task_3_result = {'A:\\Users\\Aboba\\': ['Course-Work.docx', 'Task.pdf',
'Program.zipов']}

class TasksTestCase(unittest.TestCase):
    def test_task_1(self):
        self.assertEqual(task_1_result, task_1())
    def test_task_2(self):
        self.assertEqual(task_2_result, task_2())
    def test_task_3(self):
        self.assertEqual(task_3_result, task_3())

```

## Результаты выполнения модульного тестирования:

Launching pytest with arguments C:\Users\kroll\PycharmProjects\BKIT\rk\rk2\tdd.py --no-header --no-summary -q in C:\Users\kroll\PycharmProjects\BKIT\rk\rk2

```

===== test session starts =====
collecting ... collected 3 items

```

```

tdd.py::TasksTestCase::test_task_1 PASSED [ 33%]
tdd.py::TasksTestCase::test_task_2 PASSED [ 66%]
tdd.py::TasksTestCase::test_task_3 PASSED [100%]

```

```

===== 3 passed in 0.02s =====

```

Process finished with exit code 0