Ковалев Е.А ИУ5-36Б
Вариант 15 = "Файл - Каталог файлов"
Вариант запросов – Д

Текст программы (переработанный):

```
#rk2.py

from operator import itemgetter

class File:
    def __init__(self, id, name, size, cat_id):
        self.id = id
        self.name = name # Название файла (с расширением)
        self.size = size # Размер файла (в Б)
        self.cat_id = cat_id

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

class Pair:
    def __init__(self, cat_id, file_id):
        self.cat_id = cat_id
        self.file_id = file_id

def get_all_txt_files(files, cats):
    O2M = [(f.name, f.size, c.name)
           for f in files
           for c in cats
           if f.cat_id == c.id]

    return [(file_name, cat_name)
            for file_name, file_size, cat_name in O2M
            if file_name.endswith('.txt')]

def get_cats_by_average_file_size(files, cats, pair_list):
    O2M = [(f.name, f.size, c.name)
           for f in files
           for c in cats
           if f.cat_id == c.id]

    M2M = [(f.name, f.size, cat_name)
           for cat_name, file_id in [(c.name, p.file_id)
                                     for c in cats
                                     for p in pair_list
                                     if c.id == p.cat_id]
           for f in files
```

```
          if f.id == file_id]

    res_2 = []

    for c in cats:
        c_files = list(filter(lambda i: i[2] == c.name, O2M))
        if len(c_files) > 0:
            f_size = [size for _, size, _ in c_files]
            c_size = sum(f_size)
            res_2.append((c.name, c_size / len(c_files)))

    return sorted(res_2, key=itemgetter(1), reverse=True)

def get_cats_starting_with_d(files, cats, pair_list):
    M2M = [(f.name, f.size, cat_name)
          for cat_name, file_id in [(c.name, p.file_id)
                                    for c in cats
                                    for p in pair_list
                                    if c.id == p.cat_id]
          for f in files
          if f.id == file_id]

    res_3 = {}

    for c in cats:
        if c.name.startswith('D'):
            c_files = [name for name, _, _ in list(filter(lambda i: i[2] == c.name, M2M))]
            res_3[c.name] = c_files

    return res_3

# Функция main для запуска программы
def main(files, cats, pair_list):
    print('D1 - All .txt files and cats')
    print(get_all_txt_files(files, cats))

    print('D2 - Cats by average file size')
    print(get_cats_by_average_file_size(files, cats, pair_list))

    print('D3 - All cats starting with D (and files)')
    print(get_cats_starting_with_d(files, cats, pair_list))

if __name__ == '__main__':

    cats = [
        Cat(1, 'Desktop'),
        Cat(2, 'Documents'),
        Cat(3, 'Downloads'),
        Cat(4, 'Music'),
        Cat(5, 'Pictures'),
        Cat(6, 'Videos'),
    ]
    files = [
        File(1, 'wallpaper.jpg', 3000000, 5),
        File(2, 'passwords.txt', 2048, 2),
        File(3, 'virus.exe', 20000000, 3),
        File(4, 'log.txt', 3000000, 2),
        File(5, 'readme.txt', 25000, 3),
        File(6, 'flower.jpg', 4000000, 5),
    ]

    pair_list = [
        Pair(1, 1),
        Pair(5, 1),
        Pair(2, 2),
        Pair(1, 3),
        Pair(2, 3),
        Pair(3, 3),
        Pair(4, 3),
        Pair(5, 3),
        Pair(6, 3),
        Pair(2, 4),
        Pair(2, 5),
        Pair(3, 5),
        Pair(5, 6),
    ]

    main(files, cats, pair_list)
```

Модульные тесты (с использованием unittest):

```
#tests.py

import unittest
from rk2 import *

class TestFileFunctions(unittest.TestCase):

    def setUp(self):
        self.cats = [
            Cat(1, 'Desktop'),
            Cat(2, 'Documents'),
            Cat(3, 'Downloads'),
            Cat(4, 'Music'),
            Cat(5, 'Pictures'),
            Cat(6, 'Videos'),
        ]
        self.files = [
            File(1, 'wallpaper.jpg', 3000000, 5),
            File(2, 'passwords.txt', 2048, 2),
            File(3, 'virus.exe', 20000000, 3),
            File(4, 'log.txt', 3000000, 2),
            File(5, 'readme.txt', 25000, 3),
            File(6, 'flower.jpg', 4000000, 5),
        ]
```

```python
        self.pair_list = [
            Pair(1, 1),
            Pair(5, 1),
            Pair(2, 2),
            Pair(1, 3),
            Pair(2, 3),
            Pair(3, 3),
            Pair(4, 3),
            Pair(5, 3),
            Pair(6, 3),
            Pair(2, 4),
            Pair(2, 5),
            Pair(3, 5),
            Pair(5, 6),
        ]

    def test_res1(self):
        result = get_all_txt_files(self.files, self.cats)
        expected = [('passwords.txt', 'Documents'), ('log.txt', 'Documents'), ('readme.txt', 'Downloads')]
        self.assertEqual(result, expected)

    def test_res2(self):
        result = get_cats_by_average_file_size(self.files, self.cats, self.pair_list)
        expected = [
            ('Downloads', 10012500.0),
            ('Pictures', 3500000.0),
            ('Documents', 1501024.0)]
        self.assertEqual(result, expected)

    def test_res3(self):
        result = get_cats_starting_with_d(self.files, self.cats, self.pair_list)
        expected = {
            'Desktop': ['wallpaper.jpg', 'virus.exe'],
            'Documents': ['passwords.txt', 'virus.exe', 'log.txt', 'readme.txt'],
            'Downloads': ['virus.exe', 'readme.txt']}
        self.assertEqual(result, expected)

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