

**Московский государственный технический  
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по РК№2  
Вариант №15

Выполнил:  
студент группы ИУ5-34Б  
Ступницкий Михаил  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю.Е.  
Подпись и дата:

Москва, 2023 г.

Задание:

- 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
- 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Текст программы RK1.py:

```
from operator import itemgetter

class File:

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

class Dir:

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

class FileDir:

    def __init__(self, file_id, dir_id):
        self.file_id = file_id
        self.dir_id = dir_id

directs = [
    Dir(1, 'Документы'),
    Dir(2, 'Личное'),
    Dir(3, 'Учеба'),
    Dir(4, 'Работа'),
]

files = [
    File(1, 'РК1_ПиКЯП.py', 950, 3),
    File(2, 'Хоккей_Билеты.pdf', 1838, 2),
    File(3, 'Алгоритмы_книга.pdf', 101560, 1),
    File(4, 'Отчет1.docx', 147, 4),
    File(5, 'Модель.jpg', 48, 2),
]

files_dirs = [
    FileDir(1, 1),
    FileDir(1, 3),
    FileDir(2, 1),
    FileDir(2, 2),
    FileDir(3, 1),
    FileDir(3, 3),
    FileDir(3, 4),
```

```

FileDir(4, 4),
FileDir(5, 2),
FileDir(5, 3),
FileDir(5, 4),
]

def g1_solution(one_to_many):
    names = []
    for i in range(0, len(one_to_many)):
        if one_to_many[i][2][0] == "Л":
            names.append(one_to_many[i][2])
    res1 = {name: [(otm[0], otm[1]) for otm in one_to_many if otm[2] == name]
    for name in names}
    return res1

def g2_solution(one_to_many):
    dirs = [d.name for d in directs]
    res2 = sorted([(name, max([otm[1] for otm in one_to_many if otm[2] ==
name])) for name in dirs], key=itemgetter(1),
reverse=True)
    return res2

def g3_solution(many_to_many):
    res3 = sorted(many_to_many, key=itemgetter(2))
    return [res3[i] for i in range(0, len(res3))]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(f.name, f.size, d.name)
                    for d in directs
                    for f in files
                    if f.dir_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, fd.dir_id, fd.file_id)
                          for d in directs
                          for fd in files_dirs
                          if d.id == fd.dir_id]

    many_to_many = [(f.name, f.size, dir_name)
                     for dir_name, dir_id, file_id in many_to_many_temp
                     for f in files if f.id == file_id]

    print('Задание Г1:\n«Каталог» и «Файл» связаны соотношением один-ко-
многим. Выведите список всех каталогов,
    'у которых название начинается с буквы «Л», и список хранящихся в
    них файлов.')
    print(g1_solution(one_to_many))

    print('\nЗадание Г2:\n«Каталог» и «Файл» связаны соотношением один-ко-
многим.')
    'Выведите список каталогов с максимальным размером файла в каждом
    каталоге, отсортированный по максимальному размеру файла.')
    print(g2_solution(one_to_many))

    print('\nЗадание Г3:\n«Каталог» и «Файл» связаны соотношением многие-ко-
многим.')
    'Выведите список всех связанных файлов и каталогов, отсортированный

```

```

по каталогам, сортировка по файлам произвольная. ')
    print(g3_solution(many_to_many))

if __name__ == '__main__':
    main()

```

## Текст программы RK2.py:

```

import unittest
from RK1 import *

class Test_Program(unittest.TestCase):
    directs = [
        Dir(1, 'Документы'),
        Dir(2, 'Личное'),
        Dir(3, 'Учеба'),
        Dir(4, 'Работа'),
    ]

    files = [
        File(1, 'РК1_ПиКЯП.py', 950, 3),
        File(2, 'Хоккей_Билеты.pdf', 1838, 2),
        File(3, 'Алгоритмы_книга.pdf', 101560, 1),
        File(4, 'Отчет1.docx', 147, 4),
        File(5, 'Модель.jpg', 48, 2),
    ]

    files_dirs = [
        FileDir(1, 1),
        FileDir(1, 3),
        FileDir(2, 1),
        FileDir(2, 2),
        FileDir(3, 1),
        FileDir(3, 3),
        FileDir(3, 4),
        FileDir(4, 4),
        FileDir(5, 2),
        FileDir(5, 3),
        FileDir(5, 4),
    ]

    def test_g1(self):
        one_to_many = [(f.name, f.size, d.name)
                        for d in directs
                        for f in files
                        if f.dir_id == d.id]

        self.assertEqual(g1_solution(one_to_many),
                          {'Личное': [('Хоккей_Билеты.pdf', 1838),
                                      ('Модель.jpg', 48)]})

    def test_g2(self):
        one_to_many = [(f.name, f.size, d.name)
                        for d in directs
                        for f in files
                        if f.dir_id == d.id]

        self.assertEqual(g2_solution(one_to_many),
                          [('Документы', 101560), ('Личное', 1838), ('Учеба',
950), ('Работа', 147)])

    def test_g3(self):

```

```

many_to_many_temp = [(d.name, fd.dir_id, fd.file_id)
                      for d in directs
                      for fd in files_dirs
                      if d.id == fd.dir_id]

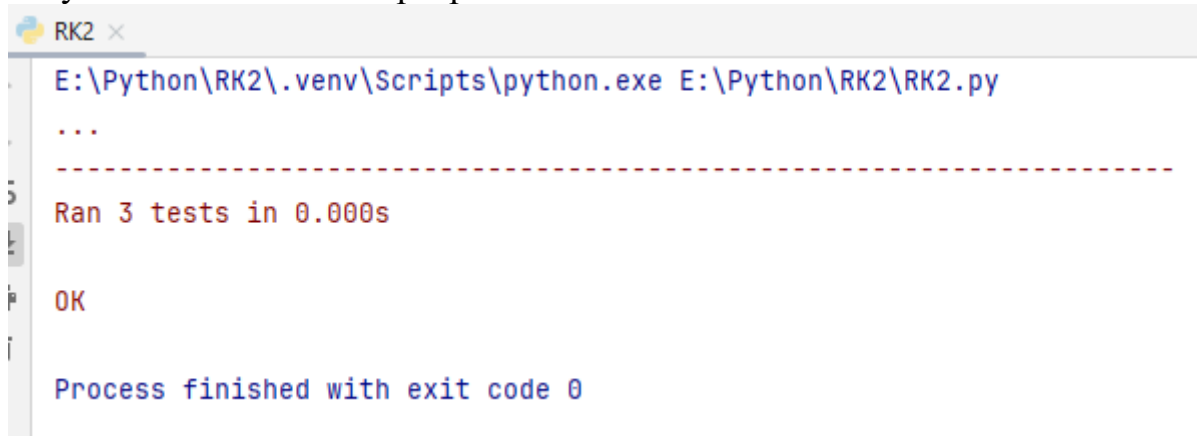
many_to_many = [(f.name, f.size, dir_name)
                 for dir_name, dir_id, file_id in many_to_many_temp
                 for f in files if f.id == file_id]

self.assertEqual(g3_solution(many_to_many),
                 [
                     ('РК1 ПиКЯП.ру', 950, 'Документы'),
                     ('Хоккей_Билеты.pdf', 1838, 'Документы'),
                     ('Алгоритмы_книга.pdf', 101560, 'Документы'),
                     ('Хоккей_Билеты.pdf', 1838, 'Личное'),
                     ('Модель.jpg', 48, 'Личное'),
                     ('Алгоритмы_книга.pdf', 101560, 'Работа'),
                     ('Отчет1.docx', 147, 'Работа'),
                     ('Модель.jpg', 48, 'Работа'),
                     ('РК1 ПиКЯП.ру', 950, 'Учеба'),
                     ('Алгоритмы_книга.pdf', 101560, 'Учеба'),
                     ('Модель.jpg', 48, 'Учеба')])

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

```

## Результат выполнения программы:



```

RK2 x
E:\Python\RK2\.venv\Scripts\python.exe E:\Python\RK2\RK2.py
...
-----
Ran 3 tests in 0.000s
OK
Process finished with exit code 0

```