

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

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

Курс «Базовые компоненты интернет-технологий»

**Отчет по Рубежному контролю №2
«Основные конструкции языка Python»**

Выполнил:
студент группы ИУ5-32Б:
Поддубный Михаил Николаевич
Подпись и дата:

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

Москва, 2021 г.

Постановка задачи:

Вариант №14

1. «Каталог файлов» и «Файл» связаны соотношением один-ко-многим. Выведите список всех связанных музыкантов и оркестров, отсортированный по файлам, сортировка по каталогам произвольная.
2. «Каталог файлов» и «Файл» связаны соотношением один-ко-многим. Выведите список каталогов с количеством файлов в каждом каталоге, отсортированный по количеству файлов.
3. «Каталог файлов» и «Файл» связаны соотношением многие-ко-многим. Выведите список всех музыкантов, у которых фамилия заканчивается на «ия», и названия их оркестров.

Условия рубежного контроля №2 по курсу БКИТ

1. Рубежный контроль представляет собой разработку тестов на языке Python.
2. 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
3. 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Текст кода:

Main.py

```
from operator import itemgetter

class Filedir:
    """Каталог файлов"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class File:
    """Файл"""
    def __init__(self, id, name, size, dir_id):
        self.id = id
        self.name = name
        self.size = size
        self.dir_id = dir_id

class FiledirFile:
    """
    'Файлы в каталоге' для реализации
    связи многие-ко-многим
    """
    def __init__(self, Filedir_id, File_id):
        self.Filedir_id = Filedir_id
```

```

        self.File_id = File_id

File_dirs = [
    Filedir(1, 'отдых'),
    Filedir(2, 'учёба'),
    Filedir(3, 'телефон'),

    Filedir(11, 'отдых (другой)'),
    Filedir(22, 'учёба (другая)'),
    Filedir(33, 'телефон (другой)'),
]

Files = [
    File(1, 'Крым', 4036, 1),
    File(2, 'Резервная копия', 349803, 3),
    File(3, 'Турция', 12089, 1),
    File(4, 'конспект', 4036, 2),
    File(5, 'программы', 9836, 3),
    File(6, 'Демидович учебник', 59836, 2),
    File(7, 'Лаба3', 2638, 2),
]

File_dir_File = [
    FiledirFile(1, 1),
    FiledirFile(1, 3),
    FiledirFile(2, 4),
    FiledirFile(3, 2),
    FiledirFile(3, 5),

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

def main():
    """Основная функция"""
    # Соединение данных один-ко-многим
    one_to_many = [(f.name, f.size, d.name)
                    for d in File_dirs
                    for f in Files
                    if f.dir_id == d.id]
    many_to_many_temp = [(d.name, fd.File_dir_id, fd.File_id)
                          for d in File_dirs
                          for fd in File_dir_File
                          if d.id == fd.File_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 B1')
    print(B1(one_to_many))
    print('Задание B2')
    print(B2(one_to_many))
    print('Задание B3')
    print(B3(many_to_many))

def B1(one_to_many):
    res_11 = sorted(one_to_many, key=lambda x: x[0].lower())
    return res_11

```

```

def B2(one_to_many):
    res_12_unsorted = []
    # Перебираем все отделы
    for d in File_dirs:
        # Список сотрудников отдела
        d_Files = list(filter(lambda i: i[2] == d.name, one_to_many))
        # Если отдел не пустой
        if len(d_Files) > 0:
            # Зарплаты сотрудников отдела
            res_12_unsorted.append((d.name, len(d_Files)))

    # Сортировка по суммарной зарплате
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    return res_12

def B3(many_to_many):
    res_13 = {}
    # Перебираем все отделы
    for f in Files:
        if f.name.endswith('ия'):
            # Список сотрудников отдела
            f_dirs = list(filter(lambda i: i[0] == f.name, many_to_many))
            # Только ФИО сотрудников
            f_dirs_names = [x for _, _, x in f_dirs]
            # Добавляем результат в словарь
            # ключ - отдел, значение - список фамилий
            res_13[f.name] = f_dirs_names
    return res_13

if __name__ == '__main__':
    main()

```

Test_TDD.py

```

import unittest
import sys, os
sys.path.append(os.getcwd())
from main import *

class Testget_B123(unittest.TestCase):
    def test_B1(self):
        one_to_many = [(f.name, f.size, d.name)
                        for d in File_dirs
                        for f in Files
                        if f.dir_id == d.id]
        self.assertEqual(B1(one_to_many), [('Демидович учебник', 59836,
'учёба'), ('конспект', 4036, 'учёба'), ('Крым', 4036, 'отдых'), ('Лаба3',
2638, 'учёба'), ('программы', 9836, 'телефон'), ('Резервная копия', 349803,
'телефон'), ('Турция', 12089, 'отдых')])

    def test_B2(self):
        one_to_many = [(f.name, f.size, d.name)
                        for d in File_dirs
                        for f in Files
                        if f.dir_id == d.id]
        self.assertEqual(B2(one_to_many), [('учёба', 3), ('отдых', 2),
('телефон', 2)])

```

```

def test_B3(self):
    many_to_many_temp = [(d.name, fd.Filedir_id, fd.File_id)
                          for d in File_dirs
                          for fd in File_dir_File
                          if d.id == fd.Filedir_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(B3(many_to_many), { 'Резервная копия' : ['телефон',
'телефон (другой)'],
                                         'Турция' : ['отдых', 'отдых
(другой)'] })

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

```

Тестирование: Main.py

```

Задание A1
('Демидович учебник', 59836, 'учёба')
('конспект', 4036, 'учёба')
('Крым', 4036, 'отдых')
('Лаба3', 2638, 'учёба')
('программы', 9836, 'телефон')
('Резервная копия', 349803, 'телефон')
('Турция', 12089, 'отдых')

Задание A2
('учёба', 3)
('отдых', 2)
('телефон', 2)

Задание A3
Резервная копия ['телефон', 'телефон (другой)']
Турция ['отдых', 'отдых (другой)']

Process finished with exit code 0

```

Test_TDD.py

```

C:\Users\vldti\PycharmProjects\RK2\venv\Scripts\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.1\plugins\python-ce\helpers\pycharm\_jb_ur
Testing started at 16:21 ...

Ran 3 tests in 0.002s

OK
Launching unittests with arguments python -m unittest C:/Users/vldti/PycharmProjects/RK2/TDD.py in C:/Users/vldti/PycharmProjects/RK2

Process finished with exit code 0

```