

**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 запросов: В  
Вариант предметной области: 25

Выполнил:  
студент группы ИУ5-33Б  
Швецов Даниил

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю. Е.

Москва, 2023 г.

## Вариант запросов В. Предметная область 25.

- 1) «Раздел» и «Документ» связаны соотношением один-ко-многим. Выведите список всех библиотек, которые начинаются с буквы «А», и названия их языков программирования.
- 2) «Раздел» и «Документ» связаны соотношением один-ко-многим. Выведите список языков программирования с минимальный размером библиотеки в каждом языке программирования, отсортированный по минимальному размеру.
- 3) «Раздел» и «Документ» связаны соотношением многие-ко-многим. Выведите список всех связанных библиотек и языков программирования, отсортированный по библиотекам, сортировка по языкам программирования произвольная.

### 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).

### Файл main.py

```
# используется для сортировки
from operator import itemgetter
from itertools import groupby

class Doc:
    """Документ"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Sec:
    """Раздел"""
    def __init__(self, id, name, leng, doc):
        self.id = id
        self.name = name
        self.leng = leng
        self.doc = doc

class SecDoc:
    """
    'Сотрудники отдела' для реализации
    связи многие-ко-многим
    """
    def __init__(self, sec_id, doc_id):
        self.doc_id = doc_id
```

```

        self.sec_id = sec_id

def OneToMany(Sec, Doc):
    return [(e.name, e.leng, d.name)
            for d in Doc
            for e in Sec
            if e.doc == d.id]

def ManyToMany(Secs, Doc, SecDoc):
    many_to_many_temp = [(d.name, ed.doc_id, ed.sec_id)
                          for d in Doc
                          for ed in SecDoc
                          if d.id==ed.doc_id]

    many_to_many = [(e.name, e.leng, doc_name)
                    for doc_name, _, dep_id in many_to_many_temp
                    for e in Secs if e.id==dep_id]
    return many_to_many

```

# Документы

```

docs = [
    Doc(1, 'Договор от 21.10.21'),
    Doc(2, 'Соглашение от 29.09.21'),
    Doc(3, 'Устав от 11.05.19'),

    Doc(6, 'Договор от 13.07.22'),
    Doc(7, 'Соглашение от 31.03.22'),
    Doc(8, 'Устав от 01.01.99'),
]

```

# Разделы

```

secs = [
    Sec(1, 'Оглавление', 1, 1),
    Sec(2, 'Автоматизация', 10, 1),
    Sec(3, 'Стоимость', 2, 2),
    Sec(4, 'Ансамбль', 5, 2),
    Sec(5, 'Условия', 7, 2),
    Sec(6, 'Предметы и виды деятельности', 3, 3),
    Sec(7, 'Участники', 11, 3),
    Sec(8, 'Уставной Капитал', 1, 8),
    Sec(9, 'преамбула', 5, 7),
    Sec(10, 'реквизиты', 6, 1),
]

```

```

secs_docs = [
    SecDoc(1,1),
    SecDoc(2,1),
    SecDoc(3,2),
    SecDoc(4,2),
    SecDoc(5,2),

```

```

SecDoc(6,3),
SecDoc(7,3),
SecDoc(8,8),
SecDoc(9,7),
SecDoc(10,1),

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

def get_first(data):
    return list(filter(lambda x: x[0].lower()[0] == 'a', data))
def get_second(data):
    grouped = groupby(data, lambda x: x[2])
    data = []
    for i in grouped:
        data.append(min(i[1], key = lambda x: x[1]))
    return sorted(data, key = lambda x: x[1])

def get_third(data):
    return sorted(data, key= lambda x: x[0])

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

    # Соединение данных один-ко-многим
    one_to_many = OneToMany(secs, docs)
    # Соединение данных многие-ко-многим
    many_to_many = ManyToMany(secs, docs, secs_docs)

    print('Задание B1')
    print("\n".join(map(lambda x: f"Документ : {x[2]}, Раздел : {x[0]}, Длинной {x[1]}",
get_first(one_to_many))))

    print("\nЗадание B2")
    print("\n".join("Документ: {2}, Раздел минимальной длины: {0}, Длина {1}".format(*i) for
i in get_second(one_to_many)))

    print("\nЗадание B3")
    print("\n".join(map(lambda x: f"Документ: {x[2]}, Раздел: {x[0]}, длина: {x[1]}",
get_third(many_to_many))))

if __name__ == '__main__':
    main()

```

Файл tests.py

```

import unittest
from main import *

class TestJoinOneToMany(unittest.TestCase):
    def test_one_to_many(self):
        result = OneToMany(secs, docs)
        self.assertTrue(len(result) == 10)

class TestJoinManyToMany(unittest.TestCase):
    def test_many_to_many(self):
        result = ManyToMany(secs, docs, secs_docs)
        self.assertTrue(len(result) == 10)

class TestResults(unittest.TestCase):
    def test_get_a1_result(self):
        data = [("Секция 1", 5, "Договор 1"), ("А Секция 2", 5, "Договор 1"), ("Секция 3", 5, "Договор 2")]
        result = get_first(data)
        self.assertEqual(result, [("А Секция 2", 5, "Договор 1")]) # Проверка корректности результата

    def test_get_a1_empty_result(self):
        data = [] # пустой список
        result = get_first(data)
        self.assertEqual(result, []) # ожидаемый результат - пустой список

    def test_get_a2_result(self):
        data = [("Секция 1", 5, "Договор 1"), ("А Секция 2", 6, "Договор 1"), ("Секция 3", 5, "Договор 2")]
        result = get_second(data)
        self.assertEqual(result, [("Секция 1", 5, "Договор 1"), ("Секция 3", 5, "Договор 2")]) # Проверка корректности результата

    def test_get_a3_result(self):
        data = [("Секция 1", 5, "Договор 1"),
                ("А Секция 2", 6, "Договор 1"),
                ("Секция 3", 5, "Договор 2")]
        result = sorted(data, key=lambda x: x[0]) # ожидаемый результат - отсортированный по имени библиотеки
        self.assertEqual(result, get_third(data))

if __name__ == '__main__':

```

## Результат выполнения

```
(base) PS C:\Personal data\projects\uni\3 sem\programming> python -u "c:\Personal data\projects\uni\3 sem\programming\rk2\main.py"
Задание B1
Документ : Договор от 21.10.21, Раздел : Автоматизация, Длинной 10
Документ : Соглашение от 29.09.21, Раздел : Ансамбль, Длинной 5

Задание B2
Документ: Договор от 21.10.21, Раздел минимальной длины: Оглавление, Длина 1
Документ: Устав от 01.01.99, Раздел минимальной длины: Уставной Капитал, Длина 1
Документ: Соглашение от 29.09.21, Раздел минимальной длины: Стоимость, Длина 2
Документ: Устав от 11.05.19, Раздел минимальной длины: Предметы и виды деятельности, Длина 3
Документ: Соглашение от 31.03.22, Раздел минимальной длины: преамбула, Длина 5

Задание B3
Документ: Договор от 21.10.21, Раздел: Автоматизация, длина: 10
Документ: Соглашение от 29.09.21, Раздел: Ансамбль, длина: 5
Документ: Договор от 21.10.21, Раздел: Оглавление, длина: 1
Документ: Устав от 11.05.19, Раздел: Предметы и виды деятельности, длина: 3
Документ: Соглашение от 29.09.21, Раздел: Стоимость, длина: 2
Документ: Соглашение от 29.09.21, Раздел: Условия, длина: 7
Документ: Устав от 01.01.99, Раздел: Уставной Капитал, длина: 1
Документ: Устав от 11.05.19, Раздел: Участники, длина: 11
Документ: Соглашение от 31.03.22, Раздел: преамбула, длина: 5
Документ: Договор от 21.10.21, Раздел: реквизиты, длина: 6
(base) PS C:\personal data\projects\uni\3 sem\programming> python -u c:\Personal data\projects\uni\3 sem\programming\rk2\tests.py" -v
test_many_to_many (__main__.TestJoinManyToMany) ... ok
test_one_to_many (__main__.TestJoinOneToMany) ... ok
test_get_a1_empty_result (__main__.TestResults) ... ok
test_get_a1_result (__main__.TestResults) ... ok
test_get_a2_result (__main__.TestResults) ... ok
test_get_a3_result (__main__.TestResults) ... ok

-----
Ran 6 tests in 0.003s

OK
```