

**Отчет по рубежному контролю №2 по курсу
базовые компоненты интернет-технологий (БКИТ)**

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-54Б
Папин Алексей

(подпись)

Гапанюк Ю.Е.

"__" _____ 2022 г.

Москва, МГТУ - 2022

СОДЕРЖАНИЕ ОТЧЕТА

1. Листинг программы:	2
1.1. RK_2.py (Основной файл)	2
1.2. test.py (Файл модульного теста).....	5
2. Результаты работы программы:	7
2.1. Основной файл (RK_2.py)	7
2.2. Модульный тест-файл (test.py)	7
2.2.1. В IDE JetBrains PyCharm	7
2.2.2. В cmd / powershell.....	8

1. Листинг программы:

1.1.RK_2.py (Основной файл)

```
from operator import itemgetter

'''
Вариант Г.
«Книга» и «Глава» связаны соотношением один-ко-многим.
Выведите список всех книг, у которых название начинается с буквы «А», и
список содержащих в них глав.

«Книга» и «Глава» связаны соотношением один-ко-многим.
Выведите список книг с максимальной страницы глав в каждой книге,
отсортированный по максимальной странице.

«Книга» и «Глава» связаны соотношением многие-ко-многим.
Выведите список всех связанных глав и книг, отсортированный по книгам,
сортировка по главам произвольная.
'''

# класс Глава
class Chapter:
    def __init__(self, id, name, page, book_id):
        # номер главы
        self.id = id
        # название главы
        self.name = name
        # страница на которой находится глава
        self.page = page
        # номер книги
        self.book_id = book_id

# класс книга
class Book:
    def __init__(self, id, name):
        self.id = id
        self.name = name

class ChapterBook:
    def __init__(self, book_id, chapter_id):
        self.book_id = book_id
        self.chapter_id = chapter_id

books = [
    Book(1, 'СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ.'),
    Book(2, 'Теоретические основы электротехники. Электрические цепи.'),
    Book(3, 'Английский язык (Для технических университетов и вузов)'),
    Book(4, 'Руководство для начинающих С++. Второе издание'),
    Book(5, 'Архитектура АСОИУ. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ')
```

```

]
chapters = [
    Chapter(1, 'Основные положения теории электромагнитного поля и их
применение к теории электрических цепей', 8, 2),
    Chapter(2, 'ОПИСАНИЕ СХЕМ', 23, 1),
    Chapter(3, 'Основы C++', 22, 4),
    Chapter(4, 'ОПИСАНИЕ СИМВОЛОВ', 139, 1),
    Chapter(5, 'Не линейные электрические цепи временного тока', 70, 2),
    Chapter(6, 'АРХИТЕКТУРНОЕ ПОНЯТИЕ МНОЖЕСТВА', 4, 5)
]
chapters_of_books = [
    ChapterBook(2, 1),
    ChapterBook(1, 2),
    ChapterBook(4, 3),
    ChapterBook(1, 4),
    ChapterBook(2, 5),
    ChapterBook(5, 6),
]

def one_to_many(books, chapters):
    return [(ch.name, ch.page, book.name)
            for book in books
            for ch in chapters
            if ch.book_id == book.id]

def many_to_many_temp(books, chapters_of_books):
    return [(book.name, ChOfBooks.book_id, ChOfBooks.chapter_id)
            for book in books
            for ChOfBooks in chapters_of_books
            if book.id == ChOfBooks.book_id]

def many_to_many(books, chapters):
    return [(ch.name, ch.page, book_name)
            for book_name, book_id, ch_id in many_to_many_temp(books,
chapters_of_books)
            for ch in chapters if ch.id == ch_id]

def exercise_G1(books, chapters):
    array_dict = {}
    for lib_name, x, book_name in one_to_many(books, chapters):
        # если название книг начинается с 'А'
        if book_name[0] == 'А':
            if book_name in array_dict:
                array_dict[book_name].append(lib_name)
            else:
                array_dict[book_name] = [lib_name]

    return array_dict.items()

def exercies_G2(books, chapters):
    array_dict_2 = {}

```

```

    for x, func_num, book_name in one_to_many(books, chapters):
        if book_name in array_dict_2:
            array_dict_2[book_name] = max(array_dict_2[book_name], func_num)
        else:
            array_dict_2[book_name] = func_num
    array_dict_2 = {key: value for key, value in sorted(array_dict_2.items(),
key=lambda item: item[1])}
    return array_dict_2.items()

def exercise_G3(books, chapters):
    array_list = []
    for lib_name, x, book_name in many_to_many(books, chapters):
        array_list.append((book_name, lib_name))
    array_list = sorted(array_list, key=lambda item: item[0])
    return array_list

def main():
    print('Задание Г1')
    print(*exercise_G1(books, chapters))

    print('Задание Г2')
    print(*exercies_G2(books, chapters))

    print('Задание Г3')
    print(*exercise_G3(books, chapters))

if __name__ == '__main__':
    main()

```

1.2.test.py (Файл модульного теста)

```
import unittest

from RK_1 import Chapter, Book, ChapterBook, exercise_G1, exercies_G2,
exercise_G3, books, chapters, chapters_of_books

simple_book = [
    Book(1, 'Азбука'),
    Book(2, 'Программирование на Python'),
    Book(3, 'Математика')
]
simple_chapter = [
    Chapter(1, 'Модульный тест', 10, 2),
    Chapter(2, 'Буква А', 1, 1),
    Chapter(3, 'Плюсик', 4, 3)
]
simple_chapter_of_books = [
    ChapterBook(1, 2),
    ChapterBook(2, 1),
    ChapterBook(3, 3)
]

# Тест на создание класс "Глава"
class test_programms(unittest.TestCase):

    # Класс Chapter с пустыми параметрами
    def test_empty_parametrs_class_chapter(self):
        with self.assertRaises(TypeError) as context:
            # Выводит ошибку, т.к. в классе не было инициализировано по
            # умолчанию
            Chapter()

        self.assertEqual(
            "__init__() missing 4 required positional arguments: 'id',
            'name', 'page', " "and 'book_id'",
            str(context.exception))

    # Класс Chapter с пустыми значениями
    def test_empty_class_chapter(self):
        myclass_chapter = Chapter(None, None, None, None)
        self.assertEqual(myclass_chapter.id, None)
        self.assertEqual(myclass_chapter.name, None)
        self.assertEqual(myclass_chapter.page, None)
        self.assertEqual(myclass_chapter.book_id, None)

    # Класс Chapter с значением
    def test_class_chapter(self):
        myclass_chapter = Chapter(1, 'Математическое моделирование', 8, 2)
        self.assertEqual(myclass_chapter.id, 1)
        self.assertEqual(myclass_chapter.name, 'Математическое
моделирование')
        self.assertEqual(myclass_chapter.page, 8)
```

```

        self.assertEqual(myclass_chapter.book_id, 2)

# Класс Book с значением
def test_class_book(self):
    myclass_book = Book(1, 'СХЕМЫ АЛГОРИТМОВ')
    self.assertEqual(myclass_book.id, 1)
    self.assertEqual(myclass_book.name, 'СХЕМЫ АЛГОРИТМОВ')

# Класс ChapterBook с значением
def test_class_chapter_book(self):
    myclass_chapter_book = ChapterBook(2, 1)
    self.assertEqual(myclass_chapter_book.book_id, 2)
    self.assertEqual(myclass_chapter_book.chapter_id, 1)

# Проверка задания Г1
def test_exercise_G1(self):
    self.assertEqual(dict(exercise_G1(simple_book, simple_chapter)),
{'Азбука': ['Буква А']})

# Проверка задания Г2
def test_exercise_G2(self):
    self.assertEqual(dict(exercies_G2(simple_book, simple_chapter)),
{'Азбука': 1, 'Математика': 4, 'Программирование на Python': 10})

# Проверка задания Г3
def test_exercise_G3(self):
    self.assertEqual(exercise_G3(simple_book, simple_chapter),
[('Азбука', 'Буква А'), ('Программирование на Python', 'Модульный тест')])

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

```

2. Результаты работы программы:

2.1. Основной файл (RK_2.py)

Задание Г1

('Архитектура АСОИУ. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ', ['АРХИТЕКТУРНОЕ ПОНЯТИЕ МНОЖЕСТВА']))

Задание Г2

('Архитектура АСОИУ. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ', 4) ('Руководство для начинающих С++. Второе издание', 22) ('Теоретические основы электротехники. Электрические цепи.', 70) ('СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ.', 139)

Задание Г3

('Архитектура АСОИУ. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ', 'АРХИТЕКТУРНОЕ ПОНЯТИЕ МНОЖЕСТВА') ('Руководство для начинающих С++. Второе издание', 'Основы С++') ('СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ.', 'ОПИСАНИЕ СХЕМ') ('СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ.', 'ОПИСАНИЕ СИМВОЛОВ') ('Теоретические основы электротехники. Электрические цепи.', 'Основные положения теории электромагнитного поля и их применение к теории электрических цепей') ('Теоретические основы электротехники. Электрические цепи.', 'Не линейные электрические цепи временного тока')

```
C:\Users\Papin\AppData\Local\Programs\Python\Python39\python.exe D:\Python\BKIT\RK_1.py
Задание Г1
('Архитектура АСОИУ. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ', ['АРХИТЕКТУРНОЕ ПОНЯТИЕ МНОЖЕСТВА']))
Задание Г2
('Архитектура АСОИУ. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ', 4) ('Руководство для начинающих С++. Второе издание', 22) ('Теоретические основы электротехники. Электрические цепи.', 70) ('СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ.', 139)
Задание Г3
('Архитектура АСОИУ. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ', 'АРХИТЕКТУРНОЕ ПОНЯТИЕ МНОЖЕСТВА') ('Руководство для начинающих С++. Второе издание', 'Основы С++') ('СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ.', 'ОПИСАНИЕ СХЕМ') ('СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ.', 'ОПИСАНИЕ СИМВОЛОВ') ('Теоретические основы электротехники. Электрические цепи.', 'Основные положения теории электромагнитного поля и их применение к теории электрических цепей') ('Теоретические основы электротехники. Электрические цепи.', 'Не линейные электрические цепи временного тока')

Process finished with exit code 0
```

Рисунок 1 - Скриншот результатов работы программы

2.2. Модульный тест-файл (test.py)

2.2.1. В IDE JetBrains PyCharm

```
Run: Python tests in test.py
Tests passed: 8 of 8 tests - 0 ms
Testing started at 23:11 ...
Launching pytest with arguments D:\Python\BKIT\test.py --no-header --no-summary -q in D:\Python\BKIT

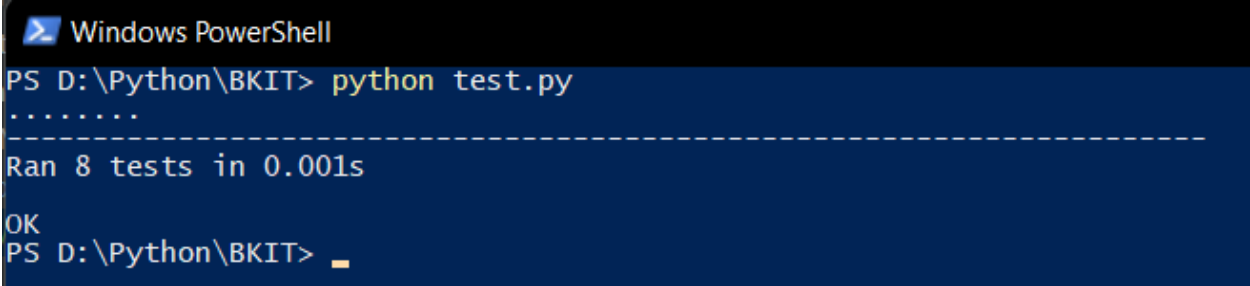
===== test session starts =====
collecting ... collected 8 items

test.py::test_programms::test_class_book PASSED [ 12%]
test.py::test_programms::test_class_chapter PASSED [ 25%]
test.py::test_programms::test_class_chapter_book PASSED [ 37%]
test.py::test_programms::test_empty_class_chapter PASSED [ 50%]
test.py::test_programms::test_empty_params_class_chapter PASSED [ 62%]
test.py::test_programms::test_exercise_G1 PASSED [ 75%]
test.py::test_programms::test_exercise_G2 PASSED [ 87%]
test.py::test_programms::test_exercise_G3 PASSED [100%]

===== 8 passed in 0.02s =====

Process finished with exit code 0
```


2.2.2. B cmd / powershell



```
Windows PowerShell
PS D:\Python\BKIT> python test.py
.....
-----
Ran 8 tests in 0.001s
OK
PS D:\Python\BKIT> 
```