

**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: 24

Выполнил:
студент группы ИУ5-33Б
Хабленко Инна

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2023 г.

Вариант запросов В. Предметная область 24.

1. «Глава» и «Книга» связаны соотношением один-ко-многим. Выведите список всех глав, начинающихся с буквы «А».
2. «Глава» и «Книга» связаны соотношением один-ко-многим. Выведите список книг с минимальным семестром, когда проходят темы этих глав, отсортированный по минимальному семестру.
3. «Глава» и «Книга» связаны соотношением многие-ко-многим. Выведите список всех связанных глав и книг, отсортированный по главам, сортировка по книгам 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).

Листинг программы.

main.py

```
from operator import itemgetter

class Chapter:
    """Глава"""
    def __init__(self, id, title, semester, book_id):
        self.id = id
        self.title = title
        self.semester = semester
        self.book_id = book_id

class Book:
    """Книга"""
    def __init__(self, id, title, author):
        self.id = id
        self.title = title
        self.author = author

class ChapterBook:
    """
    'Главы книги' для реализации связи многие-ко-многим
    """
    def __init__(self, book_id, chapter_id):
        self.book_id = book_id
        self.chapter_id = chapter_id

# Книги
books = [
    Book(1, 'Математический анализ', 'В.А.Зорич'),
    Book(2, 'Курс математического анализа', 'Л.Д.Кудрявцев'),
    Book(3, 'Сборник задач для ВТУЗОВ', 'А.В.Ефимов, Б.П.Демидович'),
    Book(11, 'Ряды и кратные интегралы', 'А.А.Гусак'),
    Book(22, 'Сборник задач по векторному анализу', 'Е.Н.Кожевников'),
    Book(33, 'Сборник задач по курсу математического анализа', 'Г.Н.Берман'),
]

# Главы
chapters = [
    Chapter(1, 'Кратные интегралы', 3, 1),
    Chapter(2, 'Анализ векторный', 3, 2),
    Chapter(3, 'Предел функции', 1, 3),
    Chapter(4, 'Дифференциалы', 2, 3),
    Chapter(5, 'Ряды', 2, 3),
]

chapters_books = [
    ChapterBook(1,1),
    ChapterBook(2,2),
    ChapterBook(3,3),
    ChapterBook(3,4),
```

```

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

# Соединение данных один-ко-многим
def one_to_many(chapters, books):
    return [(c.title, c.semester, b.title)
            for b in books
            for c in chapters
            if c.book_id==b.id]

# Соединение данных многие-ко-многим
def many_to_many(chapters, chapters_books, books):
    many_to_many_temp = [(b.title, cb.book_id, cb.chapter_id)
                          for b in books
                          for cb in chapters_books
                          if b.id==cb.book_id]
    return [(c.title, c.semester, book_title)
            for book_title, book_id, chapter_id in many_to_many_temp
            for c in chapters if c.id==chapter_id]

def get_result_1(data):
    array = sorted(data, key=itemgetter(2))
    res_11 = []
    for i in range(len(array)):
        if array[i][0].startswith("A"):
            res_11.append(array[i])
    return res_11

def get_result_2(data):
    res_12_unsorted = []
    i = 0
    for b in books:
        b_chapter = list(filter(lambda i: i[2]==b.title, data))
        if len(b_chapter) > 0:
            b_semester = [page for _,page,_ in b_chapter]
            b_semester_min = min(b_semester)
            res_12_unsorted.append((b.title, b_semester_min))
    res_12 = sorted(set(res_12_unsorted), key=itemgetter(1))
    return res_12

def get_result_3 (data):
    res_13 = sorted(data, key=itemgetter(0))
    return res_13

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

    one_to_many = one_to_many(chapters, books)
    many_to_many = many_to_many(chapters, chapters_books, books)

```

```

print('Задание B1\n')
res_11 = get_result_1(one_to_many)
for i in range(len(res_11)):
    print(*res_11[i])

print('\nЗадание B2\n')
res_12 = get_result_2(one_to_many)
for i in range(len(res_12)):
    print(*res_12[i])

print('\nЗадание B3\n')
res_13 = get_result_3(many_to_many)
for i in range(len(res_13)):
    print(*res_13[i])
print('\n')

if __name__ == '__main__':
    main()

```

tests.py

```

import unittest
from main import *

class TestJoinOperations(unittest.TestCase):
    def test_one_to_many(self):
        result = one_to_many(chapters, books)
        self.assertTrue(len(result) > 0)

    def test_many_to_many(self):
        result = many_to_many(chapters, chapters_books, books)
        self.assertTrue(len(result) > 0)

class TestTaskResults(unittest.TestCase):
    def test_get_result_1(self):
        data = [("Кратные интегралы", 3, "Математический анализ"), ("Анализ векторный", 3,
"Курс математического анализа"), ("Предел функции", 1, "Сборник задач для ВТУЗОВ")]
        result = get_result_1(data)
        self.assertEqual(result, [("Анализ векторный", 3, "Курс математического ана-
лиза")])

    def test_get_empty_result_1(self):
        data = []
        result = get_result_1(data)
        self.assertEqual(result, [])

    def test_get_result_2(self):
        data = [("Кратные интегралы", 3, "Математический анализ"), ("Анализ векторный", 3,
"Курс математического анализа"), ("Предел функции", 1, "Сборник задач для ВТУЗОВ")]
        result = get_result_2(data)
        self.assertEqual(result, [("Сборник задач для ВТУЗОВ", 1), ("Математический ана-
лиз", 3), ("Курс математического анализа", 3)])

    def test_get_result_3(self):

```

```
data = [  
    ("Анализ векторный", 3, "Сборник задач по векторному анализу1"),  
    ("Дифференциалы", 2, "Сборник задач для ВТУЗОВ"),  
    ("Дифференциалы", 2, "Сборник задач по курсу математического анализа"),  
    ("Кратные интегралы", 3, "Математический анализ"),  
    ("Кратные интегралы", 3, "Ряды и кратные интегралы"),  
    ("Предел функции", 1, "Сборник задач для ВТУЗОВ"),  
    ("Предел функции", 1, "Сборник задач по курсу математического анализа"),  
    ("Ряды", 2, "Сборник задач для ВТУЗОВ"),  
    ("Ряды", 2, "Сборник задач по курсу математического анализа")]  
result = sorted(data, key=itemgetter(0))  
self.assertEqual(result, get_result_3(data))
```

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

test_runner.py

```
import tests  
import unittest  
suite = unittest.TestLoader().loadTestsFromModule(tests)  
results = unittest.TextTestRunner(verbosity=2).run(suite)
```

Результат выполнения

```
test_many_to_many (tests.TestJoinOperations.test_many_to_many) ... ok
test_one_to_many (tests.TestJoinOperations.test_one_to_many) ... ok
test_get_empty_result_1 (tests.TestTaskResults.test_get_empty_result_1) ... ok
test_get_result_1 (tests.TestTaskResults.test_get_result_1) ... ok
test_get_result_2 (tests.TestTaskResults.test_get_result_2) ... ok
test_get_result_3 (tests.TestTaskResults.test_get_result_3) ... ok
```

```
-----
Ran 6 tests in 0.000s
```

OK