

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

**Факультет информатика и системы управления
Кафедра системы обработки информации и управления**

Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю №2

Вариант Б13

Выполнил:
студент группы ИУ5-32Б:
Угрюмов Н.А.
Подпись и дата:

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

Москва, 2025 г.

Текст программы

Файл rk1.py:

```
class Book:
    def __init__(self, id, title, author, pages, libraryID):
        self.id = id
        self.title = title
        self.author = author
        self.pages = pages
        self.libraryID = libraryID

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

class BookLibrary:
    def __init__(self, libraryID, bookID):
        self.libraryID = libraryID
        self.bookID = bookID

books = [
    Book(1, "Преступление и наказание", "Достоевский", 350, 1),
    Book(2, "Война и мир", "Толстой", 1225, 2),
    Book(3, "Мастер и Маргарита", "Булгаков", 480, 2),
    Book(4, "Евгений Онегин", "Пушкин", 320, 4),
    Book(5, "Отцы и дети", "Тургенев", 288, 3),
    Book(6, "Герой нашего времени", "Лермонтов", 224, 1),
]

libraries = [
    Library(1, "Центральная городская библиотека"),
    Library(2, "Библиотека им. Ленина"),
    Library(3, "Научная библиотека"),
    Library(4, "Детская библиотека"),
]

booksLibraries = [
    BookLibrary(1, 1),
    BookLibrary(1, 6),
    BookLibrary(2, 2),
    BookLibrary(2, 3),
    BookLibrary(3, 4),
    BookLibrary(3, 5),
    BookLibrary(4, 1),
    BookLibrary(4, 2),
    BookLibrary(4, 3),
    BookLibrary(4, 4),
]
```

```
]
```

```
def query_one_to_many(books, libraries):
    result = [
        (book.title, book.author, book.pages, lib.name)
        for book in books
        for lib in libraries
        if book.libraryID == lib.id
    ]
    return sorted(result, key=lambda x: x[0])

def query_books_count_by_library(books, libraries):
    one_to_many = query_one_to_many(books, libraries)
    result = []

    for lib in libraries:
        count = len([b for b in one_to_many if b[3] == lib.name])
        if count > 0:
            result.append((lib.name, count))

    return sorted(result, key=lambda x: x[1])

def query_many_to_many_author_ov(books, libraries, booksLibraries):
    many_to_many = [
        (book.title, book.author, lib.name)
        for bl in booksLibraries
        for book in books
        for lib in libraries
        if book.id == bl.bookID and lib.id == bl.libraryID
        and book.author.endswith("ов")
    ]

    return sorted(many_to_many, key=lambda x: x[0])
```

Файл test_rk1.py:

```
import unittest
from rk1 import (
    books, libraries, booksLibraries,
    query_one_to_many,
    query_books_count_by_library,
    query_many_to_many_author_ov
)

class TestRK1(unittest.TestCase):
    def test_query_one_to_many(self):
        result = query_one_to_many(books, libraries)
        self.assertTrue(len(result) > 0)
```

```

self.assertEqual(result[0][0], "Война и мир")

def test_query_books_count_by_library(self):
    result = query_books_count_by_library(books, libraries)
    self.assertIn("Центральная городская библиотека", 2), result)

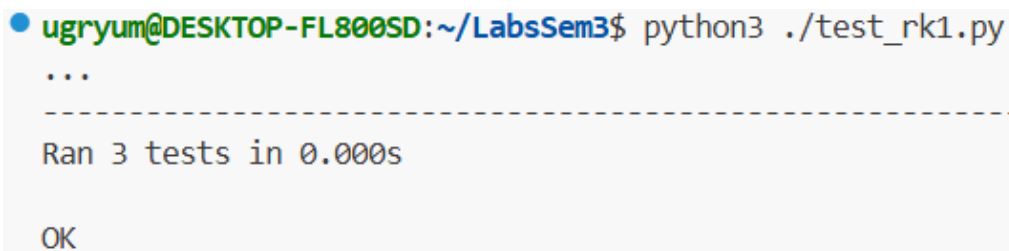
def test_query_many_to_many_author_ov(self):
    result = query_many_to_many_author_ov(books, libraries, booksLibraries)
    authors = [item[1] for item in result]

    self.assertIn("Булгаков", authors)
    self.assertIn("Лермонтов", authors)
    self.assertNotIn("Достоевский", authors)

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

```

Скриншот работы приложения



```

ugryum@DESKTOP-FL800SD:~/LabsSem3$ python3 ./test_rk1.py
...
-----
Ran 3 tests in 0.000s

OK

```

Рис. 1. вывод результатов тестов