



**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»

Отчёт по рубежному контролю №2

Выполнил:

студент группы ИУ5-32Б
Овчинников Данила Алексеевич

Подпись и дата:

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю.Е.

Подпись и дата:

Москва, 2021 г.

Описание задания

(Вариант предметной области - 12, вариант запросов - Б)

1. Провести рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
2. Для текста программы рубежного контроля №1 создать модульные тесты с применением TDD - фреймворка (3 теста).

Текст программы

Файл **main.py**:

```
from operator import itemgetter

class Book:
    """Книги"""
    def __init__(self, id, name, auth, lb_id):
        self.id = id
        self.name = name
        self.auth = auth
        self.lb_id = lb_id

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

class BooksL:
    """
    'Книги библиотеки' для реализации
    связи многие-ко-многим
    """
    def __init__(self, lb_id, book_id):
        self.lb_id = lb_id
        self.book_id = book_id

def task_1(Books, Libraries):
    # Соединение данных один-ко-многим
    one_to_many = [
        (b.name, b.auth, l.name)
        for l in Libraries
        for b in Books
        if l.id == b.lb_id
    ]
```

```

    return sorted(one_to_many, key=itemgetter(0))

def task_2(Books, Libraries):
    one_to_many = [
        (b.name, b.auth, l.name)
        for l in Libraries
        for b in Books
        if b.lb_id == l.id
    ]
    res = list()
    # Перебираем все библиотеки
    for lb in Libraries:
        # Список книг в библиотеке
        lb_books = list(filter(lambda i: i[2] == lb.name, one_to_many))
        # Если библиотека не пуста
        if len(lb_books) > 0:
            res.append((lb.name, len(lb_books)))
    return sorted(res, key=itemgetter(1), reverse=True)

def task_3(Books, Libraries, Books_Librarys):
    # Соединение данных многие-ко-многим
    many_to_many = [
        (b.name, b.auth, l.name)
        for l in Libraries
        for b in Books
        for relation in Books_Librarys
        if b.id == relation.lb_id and l.id == relation.lb_id
    ]
    res = dict()
    for b in Books:
        if b.name.endswith("ов"):
            # Ищем библиотеки, в которых есть эта книга
            b_libraries = list(filter(lambda x: x[0] == b.name, many_to_many))
            # Получаем их названия
            b_libraries_names = [x[2] for x in b_libraries]
            res[b.name] = b_libraries_names
    return res

def main():
    # Библиотеки
    Libraries = [
        Library(1, 'Ленинка'),
        Library(2, 'Питерская библиотека'),
        Library(3, 'Библиотека 11'),

        Library(11, 'Библиотека им. Макса'),
        Library(22, 'Библиотека Кубани'),
        Library(33, 'ВНДХ'),
    ]

```

```

# Книги
Books = [
    Book(1, 'Сказки', 'Артамонов', 1),
    Book(2, 'Тихое утро', 'Петров', 2),
    Book(3, 'Богатые люди', 'Иваненко', 3),
    Book(4, 'Мир и война', 'Иванов', 3),
    Book(5, 'Громкий Дон', 'Иванин', 3),
]

# Книги и библиотеки
Books_Librarys = [
    BooksL(1,1),
    BooksL(2,2),
    BooksL(3,3),
    BooksL(3,4),
    BooksL(3,5),

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

print("Задание Б1")
res_1 = task_1(Books, Libraries)
[print(e1) for e1 in res_1]

print("\nЗадание Б2")
res_2 = task_2(Books, Libraries)
[print(e1) for e1 in res_2]

print("\nЗадание Б3")
res_3 = task_3(Books, Libraries, Books_Librarys)
[print(k, v) for k, v in res_3.items()]

if __name__ == "__main__":
    main()

```

Файл **test.py**:

```

import unittest
from main import Book, Library, BooksL, task_1, task_2, task_3

class Test(unittest.TestCase):
    def setUp(self):
        self.Libraries = [
            Library(1, "Ленинка"),
            Library(2, "Питерская библиотека"),
            Library(3, "Библиотека 11"),

```

```

Library(11, "Библиотека им. Макса"),
Library(22, "Библиотека Кубани"),
Library(33, "ВНДХ"),
]
self.Books = [
    Book(1, "Сказки", "Артамонов", 1),
    Book(2, "Тихое утро", "Петров", 2),
    Book(3, "Богатые люди", "Иваненко", 3),
    Book(4, "Мир и война", "Иванов", 3),
    Book(5, "Громкий Дон", "Иванин", 3),
]
self.Books_Librarys = [
    BooksL(1,1),
    BooksL(2,2),
    BooksL(3,3),
    BooksL(3,4),
    BooksL(3,5),

    BooksL(11,1),
    BooksL(22,2),
    BooksL(33,3),
    BooksL(33,4),
    BooksL(33,5),
]
def test_task_1(self):

    key_fot_test = [
        ("Богатые люди", "Иваненко", "Библиотека 11"),
        ("Громкий Дон", "Иванин", "Библиотека 11"),
        ("Мир и война", "Иванов", "Библиотека 11"),
        ("Сказки", "Артамонов", "Ленинка"),
        ("Тихое утро", "Петров", "Питерская библиотека"),
    ]

    self.assertEqual(task_1(self.Books, self.Libraries), key_fot_test)

def test_task_2(self):

    key_fot_test = [
        ("Библиотека 11", 3),
        ("Ленинка", 1),
        ("Питерская библиотека", 1),
    ]

    self.assertEqual(task_2(self.Books, self.Libraries), key_fot_test)

def test_task_3(self):

    key_fot_test = {

```

```

        "Тихое утро": ["Питерская библиотека"],
    }

    self.assertEqual(task_3(self.Books, self.Libraries, self.Books_Librarys),
key_fot_test)

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

```

Результат выполнения программы

```

PS C:\Users\J4ngle\Desktop\rk2\code> & C:/Users/J4ngle/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/J4ngle/Desktop/rk2/code/tests.py
...
-----
Ran 3 tests in 0.000s

OK

```