



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

**Московский государственный технический университетим. Н.Э.**

**Баумана**

**(МГТУ им. Н.Э. Баумана)**

**Кафедра «Системы обработки информации и управления» (ИУ5)**

**Отчёт по рубежному контролю № 2**

**По курсу: «Базовые компоненты интернет-технологий»**

**Выполнил:**

**Садиров Фуад Этибар оглы**  
**студент группы ИУ5-31Б.**

**Проверил:**

**Гапанюк Юрий Евгеньевич**  
**Преподаватель кафедры ИУ5**

**г. Москва 2022 г.**

### 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).

### Текст программы:

#### Измененный текст программы для РК1:

```
from operator import itemgetter

class Book:
    """Книга"""

    def __init__(self, id, price, name, shop_id):
        self.id = id
        self.price = price
        self.name = name
        self.shop_id = shop_id

class Shop:
    """Книжный магазин"""

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

class BookShop:
    """ 'Книги магазина' для реализации связи многие-ко-многим """

    def __init__(self, shop_id, book_id):
        self.shop_id = shop_id
        self.book_id = book_id

Shops = [
    Shop(1, 'Книжный червь'),
    Shop(2, 'Книголюб'),
    Shop(3, 'Дом книг'),
    Shop(4, 'Мир книг'),
    Shop(5, 'Перечитай-город'),
]

Books = [
    Book(1, 68, 'Конституция Российской Федерации', 1),
    Book(2, 666, 'Божественная комедия', 2),
    Book(3, 1010, 'Совершенный алгоритм', 2),
    Book(4, 2200, 'Война и мир', 3),
    Book(5, 100, 'Сто лет одиночества', 3),
    Book(6, 1852, 'Муму', 4),
    Book(7, 1988, 'Остров сокровищ', 5)
]

Books_Shops = [
    BookShop(1, 1),
    BookShop(2, 2),
    BookShop(2, 3),
    BookShop(3, 4),
    BookShop(3, 5),
    BookShop(4, 6),
```

```

    BookShop(5, 7)
]

one_to_many = [(b.name, b.price, s.name)
               for s in Shops
               for b in Books
               if b.shop_id == s.id]

many_to_many_temp = [(s.name, bs.shop_id, bs.book_id)
                    for s in Shops
                    for bs in Books_Shops
                    if s.id == bs.shop_id]

many_to_many = [(b.id, shop_id)
                for name, shop_id, book_id in many_to_many_temp
                for b in Books if b.id == book_id]

```

### Текст программы для РК2:

```

from unittest import TestCase, main
from RK1 import *

class order_test(TestCase):
    def test_1(self):
        expected_result_for_A1 = [('Война и мир', 2200, 'Дом книг'),
                                   ('Сто лет одиночества', 100, 'Дом книг'),
                                   ('Божественная комедия', 666, 'Книголюб'),
                                   ('Совершенный алгоритм', 1010, 'Книголюб'),
                                   ('Конституция Российской Федерации', 68, 'Книжный червь'),
                                   ('Муму', 1852, 'Мир книг'),
                                   ('Остров сокровищ', 1988, 'Перечитай-город')]

        res_A1 = list(sorted(one_to_many, key=itemgetter(2)))
        self.assertEqual(res_A1, expected_result_for_A1)

    def test_2(self):
        expected_result_for_A2 = [('Книжный червь', 68), ('Книголюб', 1676),
                                   ('Мир книг', 1852), ('Перечитай-город', 1988),
                                   ('Дом книг', 2300)]

        res_A2_unsorted = []
        for s in Shops:
            s_emps = list(filter(lambda i: i[2] == s.name, one_to_many))
            if len(s_emps) > 0:
                s_prices = [price for _, price, _ in s_emps]
                s_prices_sum = sum(s_prices)
                res_A2_unsorted.append((s.name, s_prices_sum))
        res_A2 = sorted(res_A2_unsorted, key=itemgetter(1))
        self.assertEqual(res_A2, expected_result_for_A2)

    def test_3(self):
        expected_result_for_A3 = [('Дом книг', ['Война и мир', 'Сто лет одиночества']),
                                   ('Мир книг', ['Муму'])]

        res_A3 = []
        for i in filter(lambda a: "книг" in Shops[a[1] - 1].name, many_to_many):
            a = ((Shops[i[1] - 1].name, [_ .name for _ in filter(lambda a: a.shop_id == i[1],
                                                                Books)]))
            if a not in res_A3:
                res_A3.append(a)
        self.assertEqual(res_A3, expected_result_for_A3)

if __name__ == '__main__':
    main()

```

## Результаты тестирования:

✓ Tests passed: 3 of 3 tests – 2 ms

```
C:\Users\Admin\PycharmProjects\RK2\venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2022.3/plugins/python/helpers/pycharm/_jb_u
Testing started at 18:33 ...
Launching unittests with arguments python -m unittest C:\Users\Admin\PycharmProjects\RK2\test.py in C:\Users\Admin\PycharmProjects\RK2
```

Ran 3 tests in 0.003s

OK

Process finished with exit code 0