

Факультет Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

**Отчет по рубежному контролю №2 по курсу
Базовые компоненты интернет-технологий**

5

(количество листов)

Вариант № 16

Исполнитель

студент группы РТ5-316

Нижаметдинов М.Ш.

“ 21 ” декабря 2021 г.

Проверил

Доцент кафедры ИУ5

Гапанюк Ю.Е.

“ ____ ” _____ 2021 г.

Москва, 2021 г.

Задание

Рубежный контроль представляет собой разработку тестов на языке Python.

- 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

Файл «main.py»

```
# используется для сортировки
from operator import itemgetter

class Book:
    """Книга"""

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

# Книжные магазины
shops = [
    Shop(1, 'Книжный бульвар'),
    Shop(2, 'Книга инженера'),
    Shop(3, 'Книжная полка'),
    Shop(4, 'Зелёная книга'),

    Shop(11, 'Новый книжный'),
    Shop(22, 'Книгоград'),
    Shop(33, 'Книга - друг'),
]

# Книги
books = [
```

```

Book(1, 'Властелин колец', 'Джон Р. Р. Толкин', 300, 1),
Book(2, 'Адвокат Дьявола', 'Эндрю Найдерман', 350, 3),
Book(3, 'Визуализация в научных исследованиях. Учебное пособие', 'Корнеев
В.И.', 900, 2),
Book(4, '1984', 'Джордж Оруэлл', 400, 4),
Book(5, 'Тёмные начала', 'Филип Пулман', 425, 3),
Book(6, 'Гарри Поттер и Кубок огня', 'Джоан Роулинг', 340, 1),
Book(7, 'Убить пересмешника', 'Харпер Ли', 530, 3),
Book(8, 'Алая буква', 'Натаниэль Готорн', 229, 4),
]

```

```

books_shops = [
    BookShop(1, 1),
    BookShop(3, 2),
    BookShop(2, 3),
    BookShop(4, 4),
    BookShop(3, 5),
    BookShop(1, 6),
    BookShop(3, 7),
    BookShop(4, 8),

    BookShop(22, 1),
    BookShop(11, 2),
    BookShop(33, 3),
    BookShop(11, 4),
    BookShop(33, 5),
    BookShop(22, 6),
    BookShop(33, 7),
]

```

```

# Соединение данных один-ко-многим
one_to_many = [(b.name, b.author, b.price, s.name)
                for s in shops
                for b in books
                if b.shop_id == s.id]

```

```

# Соединение данных многие-ко-многим
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.name, b.author, b.price, shop_name)
                 for shop_name, shop_id, book_id in many_to_many_temp
                 for b in books
                 if b.id == book_id]

```

```

def task_1():
    res = list()
    # Перебираем все книги во всех магазинах
    for b_name, b_author, _, s_name in one_to_many:
        # Если в названии магазина содержится слово "книга",
        # добавить название этого магазина и все книги
        # этого магазина в результирующий список
        if 'книга' in s_name.lower():
            res.append((s_name, b_name, b_author))
    return res

```

```

def task_2():
    res = []
    # Перебираем все магазины
    for s in shops:
        # Получаем книги текущего магазина
        s_books = list(filter(lambda i: i[3] == s.name, one_to_many))

```

```

        if len(s_books) > 0:
            # Получаем список цен текущего магазина
            s_prices = [price for _, _, price, _ in s_books]
            # Добавляем в результирующий список имя магазина
            # и среднюю цену книги
            res.append((s.name, round(sum(s_prices) / len(s_prices), 2)))
    res = sorted(res, key=itemgetter(1))
    return res

def task_3():
    res = []
    # Перебираем все книги во всех магазинах
    for book_name, book_author, _, shop_name in many_to_many:
        # Если название книги начинается с "А",
        # в результирующий список добавляется
        # название и автор книги и все магазины,
        # в которых она продаётся
        if book_name[0] == 'А':
            res.append((book_name, book_author, shop_name))
    res = sorted(res, key=itemgetter(0))
    return res

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

    print('Задание E1')
    print(task_1())
    print('\nЗадание E2')
    print(task_2())
    print('\nЗадание E3')
    print(task_3())

if __name__ == '__main__':
    main()

```

Файл «testing.py»

```

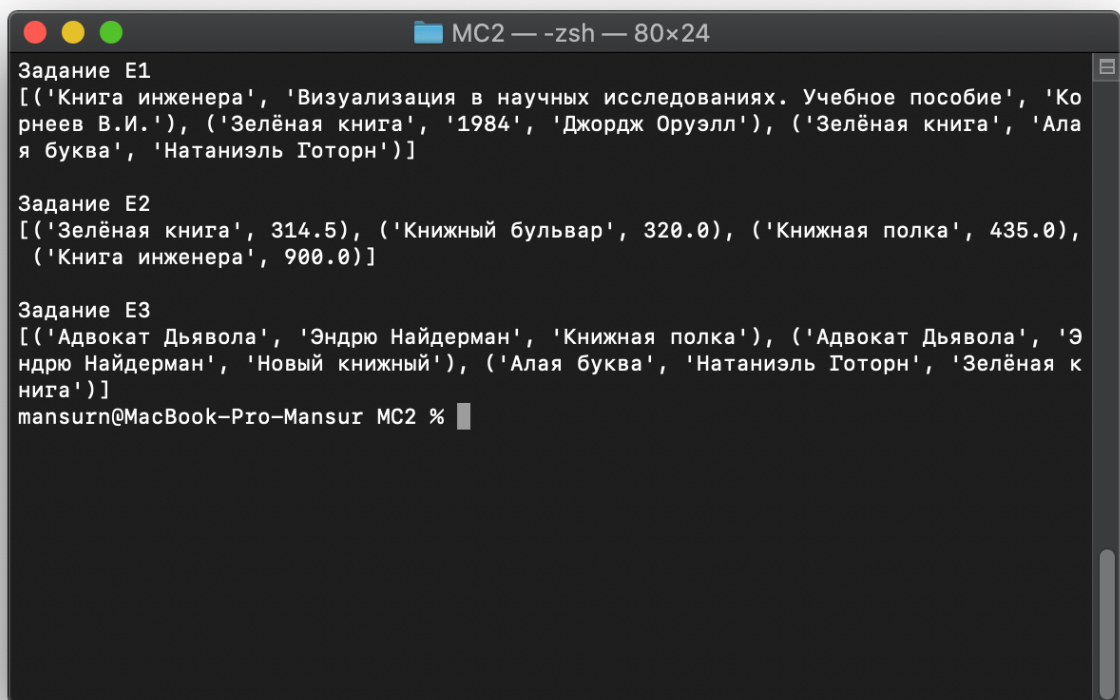
import unittest
import main

class TEST_TDD(unittest.TestCase):
    def test(self):
        self.assertEqual(main.task_1(), [('Книга инженера', 'Визуализация в научных исследованиях. Учебное пособие', 'Корнеев В.И.'), ('Зелёная книга', '1984', 'Джордж Оруэлл'), ('Зелёная книга', 'Алая буква', 'Натаниэль Готорн')])
        self.assertEqual(main.task_2(), [('Зелёная книга', 314.5), ('Книжный бульвар', 320.0), ('Книжная полка', 435.0), ('Книга инженера', 900.0)])
        self.assertEqual(main.task_3(), [('Адвокат Дьявола', 'Эндрю Найдерман', 'Книжная полка'), ('Адвокат Дьявола', 'Эндрю Найдерман', 'Новый книжный'), ('Алая буква', 'Натаниэль Готорн', 'Зелёная книга')])

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

```

Экранные формы с примерами выполнения программы

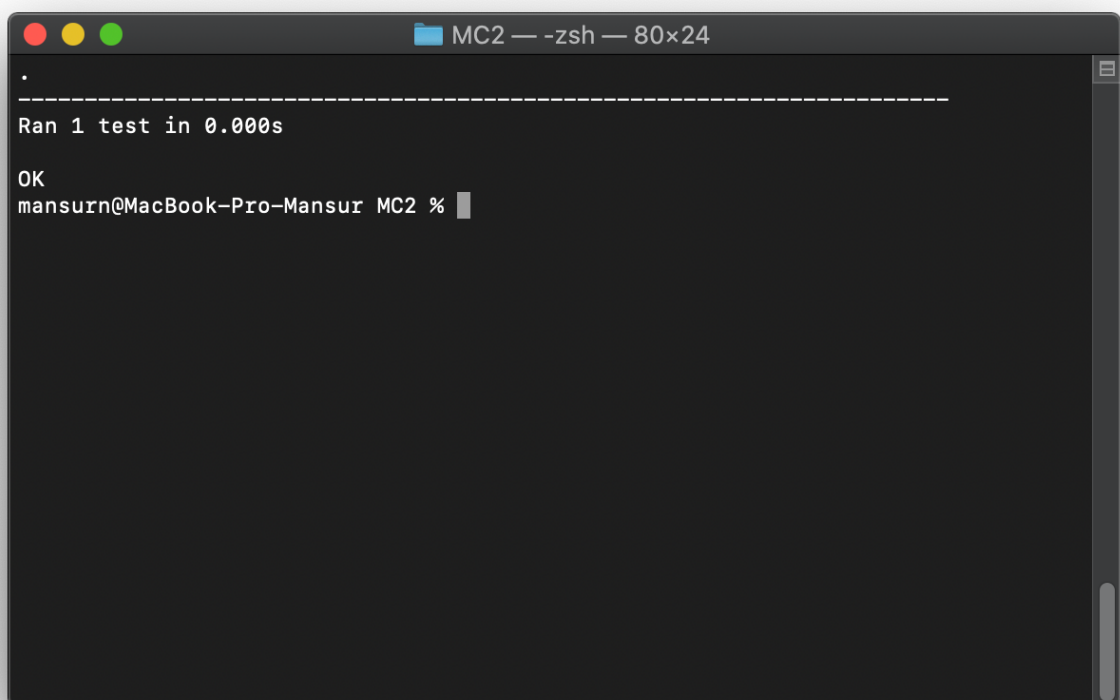


```
MC2 — -zsh — 80x24

Задание E1
[('Книга инженера', 'Визуализация в научных исследованиях. Учебное пособие', 'Корнеев В.И.'), ('Зелёная книга', '1984', 'Джордж Оруэлл'), ('Зелёная книга', 'Алая буква', 'Натаниэль Готорн')]

Задание E2
[('Зелёная книга', 314.5), ('Книжный бульвар', 320.0), ('Книжная полка', 435.0), ('Книга инженера', 900.0)]

Задание E3
[('Адвокат Дьявола', 'Эндрю Найдерман', 'Книжная полка'), ('Адвокат Дьявола', 'Эндрю Найдерман', 'Новый книжный'), ('Алая буква', 'Натаниэль Готорн', 'Зелёная книга')]
mansurn@MacBook-Pro-Mansur MC2 %
```



```
MC2 — -zsh — 80x24

.
-----
Ran 1 test in 0.000s

OK
mansurn@MacBook-Pro-Mansur MC2 %
```