Московский государственный технический

университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Программирование на основе классов и шаблонов»

Отчет по рубежному контролю №2

Вариант №16

Выполнил:

студент группы ИУ5-24Б

Соколов Б. О.

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

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).

Код:

main.py

```
# используется для сортировки
from operator import itemgetter


class Book:
    """Книга"""

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


class Shop:
    """Магазин"""

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


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

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



# Магазины
Shops = [
    Shop(1, 'Книжки для детей'),
    Shop(4, 'Книги для подростков'),
```

```
    Shop(8, 'Книжищи совсем не для детей'),
    Shop(12, 'А это книги для пожилых'),
]

# Книги
Books = [
    Book(1, 'Охота и рыбалка', 500, 4),
    Book(5, 'Приключения Вовы', 250, 1),
    Book(10, 'Миша Угрюмов и Английский, история прогула', 750, 8),
    Book(15, 'Петя и волк', 100, 1),
    Book(20, '60 цветков фиалетового', 800, 8),
    Book(25, 'Здорово жить здорОво =)', 350, 12),
]

Books_Shops = [
    BookShop(1, 1),
    BookShop(1, 4),
    BookShop(1, 12),
    BookShop(5, 1),
    BookShop(5, 4),
    BookShop(10, 1),
    BookShop(10, 8),
    BookShop(20, 12),
    BookShop(25, 12),
    BookShop(25, 8),
]

# Соединение данных один-ко-многим
one_to_many = [(b.name, b.pr, s.name_s)
               for b in Books
               for s in Shops
               if b.shop_id == s.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(s.name_s, bs.book_id, bs.shop_id)
                     for s in Shops
                     for bs in Books_Shops
                     if s.id == bs.shop_id]

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

def G1 (one_to_many):
    print('Задание Г1')
    first = {}
    for shp in Shops:
```

```
        if shp.name_s[0] == 'A':
            s_books = list(filter(lambda i: i[2] == shp.name_s, one_to_many))
            s_books_names = [x for x, _, _ in s_books]
            first[shp.name_s] = s_books_names
    return first

def G2 (one_to_many):
    print('\nЗадание Г2')
    second = []
    for shp in Shops:
        s_books = list(filter(lambda i: i[2] == shp.name_s, one_to_many))
        if len(s_books) > 0:
            books_prices = [pr for _,pr,_ in s_books]
            max_price = max(books_prices)
            second.append((shp.name_s, max_price))
    second_f = sorted(second, key=itemgetter(1), reverse=True)
    return second_f

def G3 (many_to_many):
    print('\nЗадание Г3')
    third = sorted(many_to_many, key=itemgetter(2))
    return third
```

tests.py

```
import unittest as UT
from main import *

class Testirovanie (UT.TestCase):
    def test_G1(self):
        self.assertEqual(G1(one_to_many), {'А это книги для пожилых': ['Здорово жить здорОво =)']})

    def test_G2(self):
        self.assertEqual(G2(one_to_many), [('Книжищи совсем не для детей', 800), ('Книги для подростков', 500), ('А это книги для пожилых', 350), ('Книжки для детей', 250)])

    def test_G3(self):
        self.assertEqual(G3(many_to_many), [('Охота и рыбалка', 500, 'А это книги для пожилых'),
                                            ('60 цветков фиалетового', 800, 'А это книги для пожилых'),
                                            ('Здорово жить здорОво =)', 350, 'А это книги для пожилых'),
```

```
                                        ('Охота и рыбалка', 500, 'Книги для
подростков'),
                                        ('Приключения Вовы', 250, 'Книги для
подростков'),
                                        ('Миша Угрюмов и Английский, история
прогула', 750, 'Книжищи совсем не для детей'),
                                        ('Здорово жить здорОво =)', 350, 'Книжищи
совсем не для детей'),
                                        ('Охота и рыбалка', 500, 'Книжки для
детей'),
                                        ('Приключения Вовы', 250, 'Книжки для
детей'),
                                        ('Миша Угрюмов и Английский, история
прогула', 750, 'Книжки для детей')])

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

Работа:

```
Задание Г1
.
Задание Г2
.
Задание Г3
.
----------------------------------------------------------------------
Ran 3 tests in 0.001s

OK
PS Z:\Сливки\Pytonchik\Рк 2>
```