

**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»

Отчет по Рубежному контролю №2  
13 вариант

Выполнил:  
студент группы ИУ5-32Б:  
Лапотко Ирина Евгеньевна  
Подпись и дата:

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

Москва, 2022 г.

## Постановка задачи:

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:

### RK2.py

```
from operator import itemgetter

class Book :
    def __init__(self, ID, name, num, rack_ID):
        self.ID = ID
        self.name = name
        self.num = num
        self.rack_ID = rack_ID

class Library:
    def __init__(self, ID, rack):
        self.ID = ID
        self.rack = rack

class BookLibrary:
    def __init__(self, rack_ID, book_ID):
        self.book_ID = book_ID
        self.rack_ID = rack_ID

Books = [
    Book(1, 'Анна Каренина', 4, 1),
    Book(2, 'Война и Мир', 10, 1),
    Book(3, 'Мертвые души', 3, 2),
    Book(4, 'Преступление и наказание', 3, 2),
    Book(5, 'Унесенные Ветром', 5, 3),
    Book(6, 'Горе от ума', 6, 4)
]

Lib = [
    Library(1, ' Л. Н. Толстой'),
    Library(2, 'Русская проза'),
    Library(3, 'Зарубежная проза'),
    Library(4, 'Пьеса')
]
```

```

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

def sort_name(mas):
    return sorted(mas, key = itemgetter(0))

def book_in_lib(mas, Lib2):
    res2 = []
    for l in Lib2:
        books Rack = list(filter(lambda i: i[2] == l.rack, mas))
        rack_num = len(books Rack)
        res2.append((l.rack, rack_num))
    return res2

def run_book(mas, Lib3):
    res3 = []
    for l in Lib3:
        books Rack = list(filter(lambda i: i[2] == l.rack, mas))
        for bn in books Rack:
            if 'np' in bn[0]:
                res3.append((l.rack, bn[0]))
    return res3

def main():

    one_to_many = [(b.name, b.num, l.rack)
        for b in Books
        for l in Lib
        if b.rack_ID == l.ID]

    many_to_many_temp = [(l.rack, bl.rack_ID, bl.book_ID)
        for l in Lib
        for bl in books_lib
        if l.ID == bl.rack_ID]

    many_to_many = [(b.name, b.num, lib_rack)
        for lib_rack, rack_ID, book_ID in many_to_many_temp
        for b in Books if b.ID == book_ID]

    #Сортировка по названию книги
    print("Задание Б1")
    #res1 = sorted(one_to_many, key = itemgetter(0))
    res1 = sort_name(one_to_many)
    print(res1)

```

```
#Какие книги есть на стеллажах
print("\nЗадание Б2")
res2 = book_in_lib(one_to_many, Lib)
print(res2)

#Нахождение книги по окончанию
print("\nЗадание Б3")
res3 = run_book(many_to_many, Lib)
print(res3)

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

**testsRK2.py**[illegible]

```

        if l.ID == bl.rack_ID]

    self.many_to_many = [(b.name, b.num, lib_rack)
        for lib_rack, rack_ID, book_ID in self.many_to_many_temp
        for b in self.Books if b.ID == book_ID]

    def testSortname(self):
        res = sort_name(self.one_to_many)
        res2 = [('Анна Каренина', 4, ' Л. Н. Толстой'),
            ('Война и Мир', 10, ' Л. Н. Толстой'),
            ('Горе от ума', 6, 'Пьеса'),
            ('Мертвые души', 3, 'Русская проза'),
            ('Преступление и наказание', 3, 'Русская проза'),
            ('Унесенные Ветром', 5, 'Зарубежная проза')]
        self.assertEqual(res,res2)

    def test_book_in_lib(self):
        res = book_in_lib(self.one_to_many, self.Lib)
        res2 = [(' Л. Н. Толстой', 2), ('Русская проза', 2), ('Зарубежная проза',
1), ('Пьеса', 1)]
        self.assertEqual(res,res2)

    def test_run_book(self):
        res = run_book(self.many_to_many, self.Lib)
        res2 = [(' Л. Н. Толстой', 'Война и Мир')]
        self.assertEqual(res,res2)

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

```

## Экранные формы с примерами выполнения программы

```

-----
Ran 3 tests in 0.001s

```

```

OK

```

```

PS C:\Users\user\Desktop\ИУ5\БКИТ_ЛР\RK2> 

```