

**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

Выполнил:

студент группы ИУ5-34Б
Александр Федотов

Подпись и дата:

13.12.21

Проверил:

Подпись и дата:

Москва, 2021г.

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).

Файл class_1

```
"""
класс_1
"""
class Book:

    def __init__(self, book_id, name, lists, num_shop):
        self.name_book = name
        self.lists = lists
        self.num_shop = num_shop
        self.book_id = book_id

    def __repr__(self):
        return ""
```

Файл class_2

```
"""
класс_2
"""
class Book_shop:

    def __init__(self, id_sop, name):
        self.id_shop = id_sop
        self.name = name

    def __repr__(self):
        return "Оркест {} имеет номер {}".format(self.name, self.num_orch)
```

Файл class_3

```
"""
Связующее звено
"""
class book_and_shop:

    def __init__(self, num_shop_ID, book_id_of_shop):
        self.book_id_Sh_ = book_id_of_shop
        self.num_shop_ID = num_shop_ID
```

Файл main

```
from classes.class_1 import *
from classes.class_2 import *
from classes.class_3 import *
"""
Задание_1 Выведите список всех магазинов, у которых название начинается с
буквы «А», и список книг в них.
```

```

"""
"""
Задание_2
Вывод списка магазинов с максимальной численностью страниц в книгах которые
присутствуют в данных магазинах
"""
"""
Задание_3
Вывод всех связанных магазинов и книг в них отсортированный по магазинам а так
же по количеству книг в каждом магазине
"""

books = [
    Book(1, "Книга_1", 2500, 1),
    Book(2, "А_Книга_2", 250, 2),
    Book(3, "Б_Книга_3", 50, 3),
    Book(4, "Л_Книга_4", 10, 4),
    Book(5, "и_Книга_5", 150, 3),
    Book(6, "х_Книга_6", 200, 4),
    Book(7, "Книга_7", 350, 2),
    Book(8, "Книга_8", 455, 1),
    Book(9, "Книга_9", 55, 4),
    Book(10, "Книга_10", 123, 1),
    Book(11, "Книга_11", 226, 2),
    Book(12, "Книга_12", 104, 3)
]

shops = [
    Book_shop(1, "А это магазин_1"),
    Book_shop(2, "Магазин_2"),
    Book_shop(3, "Супер магазин агазин_3"),
    Book_shop(4, "Дупер пупер магазин_4")
]

Shops_and_books = [
    book_and_shop(1, 2),
    book_and_shop(1, 3),
    book_and_shop(1, 5),
    book_and_shop(2, 2),
    book_and_shop(2, 3),
    book_and_shop(2, 1),
    book_and_shop(3, 1),
    book_and_shop(3, 2),
    book_and_shop(3, 6),
    book_and_shop(3, 3),
    book_and_shop(4, 1),
    book_and_shop(4, 2),
    book_and_shop(4, 4),
    book_and_shop(4, 3)
]

def task_1(one_to_many):
    ans = []
    for i in range(len(one_to_many)):
        if((one_to_many[i][2])[0] == 'A'):
            ans.append(one_to_many[i])
    return ans

def task_2(one_to_many):
    a = True
    ans_2 = []
    while(a):
        tmp_name = one_to_many[0][2]
        tmp_max = 0
        tmp_i = 0
        tmp_max_money = 0
        id_start = 0
        id_stop = 0

```

```

        for i in range(len(one_to_many)):
            if (one_to_many[i][2] == tmp_name and one_to_many[i][1] >
tmp_max_money):
                tmp_max_money = one_to_many[i][1]
                tmp_i = i
        for i in range(len(one_to_many)):
            if (one_to_many[i][2] == tmp_name):
                id_stop = i

        ans_2.append(one_to_many[tmp_i])

        del one_to_many[id_start:id_stop+1]
        if (len(one_to_many) == 0):
            a = False
    for i in range(len(ans_2)):
        print(ans_2[i])
    return ans_2
def task_3(man_to_many):
    a = True
    ans_3 = []

    while(a):
        tmp_name = man_to_many[0][2]
        id_start = 0
        id_stop = 0
        for i in range(len(man_to_many)-1):
            if (man_to_many[i][2] == man_to_many[i+1][2]):
                if (man_to_many[i][1] > man_to_many[i+1][1]):
                    tmp = man_to_many[i]
                    man_to_many[i] = man_to_many[i+1]
                    man_to_many[i+1] = tmp
        for i in range(len(man_to_many)):
            if (man_to_many[i][2] == tmp_name):
                id_stop = i
                ans_3.append(man_to_many[i])
        del man_to_many[id_start:id_stop + 1]
        if (len(man_to_many) == 0):
            a = False

    return ans_3

def main():
    one_to_many = [
        (b.name_book, b.lists, bs.name)
        for bs in shops
        for b in books
        if bs.id_shop == b.num_shop
    ]
    man_to_many = [
        (b.name_book, b.lists, bs.name)
        for bs in shops
        for b in books
        for rl in Shops_and_books
        if bs.id_shop == rl.num_shop_ID and b.book_id == rl.book_id_Sh_
    ]

    print("Задание_1")
    ans = task_1(one_to_many)
    for i in range(len(ans)):
        print(ans[i])
    print("Задание_2")
    ann = task_2(one_to_many)
    for i in range(len(ann)):

```

```

        print(ann[i])

    print("Задание 3")
    ans_3 = task_3(man_to_many)
    for i in range(len(ans_3)-1):
        print(ans_3[i], end= ' ')
        if(ans_3[i][2] != ans_3[i+1][2]):
            print(' ')
    print(ans_3[len(ans_3)-1],end=' ')

if __name__ == '__main__':
    main()

```

Файл TDD

```

import unittest
from main import task_1,task_2,task_3
from classes.class_1 import *
from classes.class_2 import *
from classes.class_3 import *

books = [
    Book(1,"Книга_1",2500,1),
    Book(2,"А_Книга_2",250,2),
    Book(3,"Б_Книга_3",50,3),
    Book(4,"Л_Книга_4",10,4),
    Book(5,"и_Книга_5",150,3),
    Book(6,"х_Книга_6",200,4),
    Book(7,"Книга_7",350,2),
    Book(8,"Книга_8",455,1),
    Book(9,"Книга_9",55,4),
    Book(10,"Книга_10",123,1),
    Book(11,"Книга_11",226,2),
    Book(12,"Книга_12",104,3)
]

shops = [
    Book_shop(1,"А это магазин_1"),
    Book_shop(2,"Магазин_2"),
    Book_shop(3,"Супер магазин агазин_3"),
    Book_shop(4,"Дупер пупер магазин_4")
]

Shops_and_books = [
    book_and_shop(1,2),
    book_and_shop(1,3),
    book_and_shop(1,5),
    book_and_shop(2,2),
    book_and_shop(2,3),
    book_and_shop(2,1),
    book_and_shop(3,1),
    book_and_shop(3,2),
    book_and_shop(3,6),
    book_and_shop(3,3),
    book_and_shop(4,1),
    book_and_shop(4,2),
    book_and_shop(4,4),
    book_and_shop(4,3)
]

one_to_many = [
    (b.name_book, b.lists, bs.name)
    for bs in shops
    for b in books

```

```

        if bs.id_shop == b.num_shop
    ]
    man_to_many = [
        (b.name_book, b.lists, bs.name)
        for bs in shops
        for b in books
        for rl in Shops_and_books
        if bs.id_shop == rl.num_shop_ID and b.book_id == rl.book_id_Sh_
    ]

class TestTDD(unittest.TestCase):
    def test1(self):
        self.assertEqual(task_1(one_to_many), [('Книга_1', 2500, 'А это
магазин_1'), ('Книга_8', 455, 'А это магазин_1'), ('Книга_10', 123, 'А это
магазин_1')])
    def test2(self):
        self.assertEqual(task_2(one_to_many), [('Книга_1', 2500, 'А это
магазин_1'), ('Книга_7', 350, 'Магазин_2'),
                                                ('и_Книга_5', 150, 'Супер магазин
агазин_3'),
                                                ('х_Книга_6', 200, 'Дупер пупер
магазин_4')])
    def test3(self):
        self.assertEqual(task_3(man_to_many), [('Б_Книга_3', 50, 'А это
магазин_1'), ('и_Книга_5', 150, 'А это магазин_1'), ('А_Книга_2', 250, 'А это
магазин_1'), ('Б_Книга_3', 50, 'Магазин_2'), ('А_Книга_2', 250,
'Магазин_2'), ('Книга_1', 2500, 'Магазин_2'), ('Б_Книга_3', 50, 'Супер магазин
агазин_3'), ('х_Книга_6', 200, 'Супер магазин агазин_3'), ('А_Книга_2', 250,
'Супер магазин агазин_3'), ('Книга_1', 2500, 'Супер магазин
агазин_3'), ('Л_Книга_4', 10, 'Дупер пупер магазин_4'), ('Б_Книга_3', 50,
'Дупер пупер магазин_4'), ('А_Книга_2', 250, 'Дупер пупер
магазин_4'), ('Книга_1', 2500, 'Дупер пупер магазин_4')])

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

```

Результат тестов

```

"D:\Лабы и прога 3 сем\RK_1\Scripts\python.exe" "C:\Program Files\JetBrains\PyCharm 2021.2.1\plugins\python\helpers\pycharm\_jb_unittest_runner.py"
Testing started at 12:17 ...

Ran 3 tests in 0.002s

OK
Launching unittests with arguments python -m unittest D:\Лабы и прога 3 сем\RK_1\TDD.py in D:\Лабы и прога 3 сем\RK_1

```

```

Ran 3 tests in 0.002s

OK
Launching unittests with argu

```