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
Свечникова Анна
Подпись и дата:

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)

Текст программы:

Файл library.py

```
class Library:
    #библиотека
    def __init__(self, id, addr):
        self.id = id #id библиотеки
        self.addr = addr #адрес библиотеки
```

Файл book.py

```
class Book:
    #книга
    def __init__(self, id, title, year, lib_id):
        self.id = id #id книги
        self.title = title #название книги
        self.year = year #год издания
        self.lib_id = lib_id #id библиотеки
```

Файл bookinlib.py

```
class BookInLib:
    #'книги в библиотеке' - для связи многие-ко-многим
    def __init__(self, bk_id, lib_id):
        self.bk_id = bk_id
        self.lib_id = lib_id
```

Файл data.py

```
from library import Library
from book import Book
from bookinlib import BookInLib
#список библиотек
librs = [
    Library(1, 'Алтайская улица, 4'),
    Library(2, 'Цветной бульвар, 2'),
    Library(3, 'Авиамоторная улица, 8'),
]

#список книг
books = [
    Book(1, 'Лунный камень', 1868, 1),
    Book(2, 'Шум и ярость', 1929, 2),
    Book(3, 'Гордость и предубеждение', 1813, 3),
```

```
    Book(4, 'Портрет Дориана Грея', 1890, 1),
    Book(5, 'Хлеб по водам', 1981, 3),
]

#список связей многие-ко-многим
bk_in_lbs = [
    BookInLib(1, 1),
    BookInLib(1, 2),
    BookInLib(2, 3),
    BookInLib(3, 1),
    BookInLib(4, 1),
    BookInLib(5, 2),
]
# связь один-ко-многим
one_many = [(l.addr, b.title, b.year)
            for l in librs
            for b in books
            if l.id == b.lib_id]

# свяжем названия книг и id библиотек, в которых они есть,
# на основе элементов списка bk_in_lbs
many_many_temp = [(b.title, e.lib_id)
              for b in books
              for e in bk_in_lbs
              if b.id == e.bk_id]
# теперь вместо id библиотек подставим их адреса
many_many = [(i[0], l.addr)
          for i in many_many_temp
          for l in librs
          if l.id == i[1]]
```

Файл tests.py

```
# входные данные
from data import *
# тестируемые функции
from main import task1, task2, task3
import unittest

class Tests(unittest.TestCase):

    def test_task1(self):
        self.assertEqual(task1(one_many), {'Алтайская улица, 4': [('Лунный камень', 1868),
                                                  ('Портрет Дориана Грея', 1890)], 'Авиамоторная улица,
                                                   8': [('Гордость и предубеждение', 1813), ('Хлеб по
                                                   водам', 1981)]})

    def test_task2(self):
        self.assertEqual(task2(one_many), [('Авиамоторная улица, 8', 1981),
                                           ('Цветной бульвар, 2', 1929), ('Алтайская улица, 4',1890)])
```

```
    def test_task3(self):
        self.assertEqual(task3(many_many), [('Лунный камень', 'Цветной бульвар, 2'), ('Хлеб по
                                                водам', 'Цветной бульвар, 2'),
                                                ('Лунный камень', 'Алтайская улица, 4'), ('Гордость
                                                и предубеждение', 'Алтайская улица, 4'),
                                                ('Портрет Дориана Грея', 'Алтайская улица, 4'),
                                                ('Шум и ярость', 'Авиамоторная улица, 8')])


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

Файл main.py

```
from operator import itemgetter
from data import *


def task1(src):
    """Г1 - список всех библиотек, адрес которых начинается с буквы А, и список книг в них"""
    # отбираем удовлетворяющие условию библиотеки
    chosen_lbs = list(filter(lambda l: l.addr[0:1] == 'A', librs))
    tsk_1 = {}
    # для каждой выбранной библиотеки формируем список книг, которые в ней есть
    for l in chosen_lbs:
        tsk_1[l.addr] = list((i[1], i[2]) for i in src if i[0] == l.addr)
    return tsk_1


def task2(src):
    """Г2 - список библиотек с максимальным годом издания книги
       в каждой библиотеке, отсортированный по максимальному году"""
    tsk_2_unsorted = []  # вспомогательный результирующий список
    for l in librs:
        # список годов издания книг в данной библиотеке
        l_yrs = list(i[2] for i in src if i[0] == l.addr)
        if len(l_yrs) > 0:
            # найдём максимальный год издания
            m_year = max(l_yrs)
            # добавим пару библиотека-год к результирующему списку
            tsk_2_unsorted.append((l.addr, m_year))
    # отсортируем результирующий список
    tsk_2 = sorted(tsk_2_unsorted, key=itemgetter(1), reverse=True)
    return tsk_2


def task3(src):
    """Г3 - вывести список всех связанных книг и библиотек,
        отсортированный по библиотекам (связь многие-ко-многим)"""
    tsk_3 = sorted(src, key=itemgetter(1), reverse=True)
```

```
    return tsk_3


def main():
    # основная функция
    # связь один-ко-многим
      """Г1 - список всех библиотек, адрес которых начинается с буквы А, и список книг в них"""
    print('Задание Г1')
    print(task1(one_many))

    """Г2 - список библиотек с максимальным годом издания книги
    в каждой библиотеке, отсортированный по максимальному году"""
    print('\nЗадание Г2')
    print(task2(one_many))

    """Г3 - вывести список всех связанных книг и библиотек,
    отсортированный по библиотекам (связь многие-ко-многим)"""
    print('\nЗадание Г3')

    print(task3(many_many))


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

Результат выполнения

```
anna@anna-UX310UAK:~$ /usr/local/bin/python3 /home/anna/rk.py
Задание Г1
{'Алтайская улица, 4': [('Лунный камень', 1868), ('Портрет Дориана Грея', 1890)],
 'Авиамоторная улица, 8': [('Гордость и предубеждение', 1813), ('Хлеб по водам',
1981)]}
Задание Г2
[('Авиамоторная улица, 8', 1981), ('Цветной бульвар, 2', 1929), ('Алтайская улица
, 4', 1890)]

Задание Г3
[('Лунный камень', 'Цветной бульвар, 2'), ('Хлеб по водам', 'Цветной бульвар, 2')
, ('Лунный камень', 'Алтайская улица, 4'), ('Гордость и предубеждение', 'Алтайска
я улица, 4'), ('Портрет Дориана Грея', 'Алтайская улица, 4'), ('Шум и ярость', 'А
виамоторная улица, 8')]
anna@anna-UX310UAK:~$
3.9.7 64-bit  ⊗ 0 ⚠ 0   Ln 97, Col 9  Spaces: 4  UTF-8  LF  Python
```

Результат тестирования

```
/home/anna/PycharmProjects/rk2/venv/bin/python /snap/pycharm-professional/265/plugins/python/helpers/pycharm/_jb_unittest_runner.py --p
Testing started at 21:32 ...
Launching unittests with arguments python -m unittest /home/anna/PycharmProjects/rk2/tests.py in /home/anna/PycharmProjects/rk2


Process finished with exit code 0


Ran 3 tests in 0.003s

OK
```