

Московский Государственный Технический Университет им. Баумана

Отчет по рубежному контролю № 2
По дисциплине «Базовые компоненты интернет-технологий»
Вариант № 16Б

Выполнила:
Студентка группы ИУ5-32Б
Лялько Никита
28.12.2022

Проверил:
Преподаватель кафедры ИУ5
Гапанюк Ю. Е.

Москва, 2022 г.

Задание для РК2:

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:

main.py

```
main.py x testing.py x
1 # используется для сортировки
2 from operator import itemgetter
3
4
5 class Book:
6     """Книга"""
7
8     def __init__(self, id, name, price, shop_id):
9         self.id = id
10        self.name = name
11        self.price = price
12        self.shop_id = shop_id
13
14
15 class Shop:
16     """Книжный магазин"""
17
18     def __init__(self, id, name):
19         self.id = id
20         self.name = name
21
22
23 class BookShop:
24     """
25     'Книги магазина' для реализации
26     связи многие-ко-многим
27     """
28
29     def __init__(self, shop_id, book_id):
30         self.shop_id = shop_id
31         self.book_id = book_id
32
33
34 # Магазины
35 shops = [
36     Shop(1, 'Весёлая книжка'),
37     Shop(2, 'Читатель'),
38     Shop(3, 'Книжный'),
39
40     Shop(11, 'Весёлая книжка (другой)'),
41     Shop(22, 'Читатель (другой)'),
42     Shop(33, 'Книжный (другой)'),
43 ]
44
45 # Книги
46 books = [
47     Book(1, 'Гарри Поттер', 250, 1),
48     Book(2, 'Незнайка', 300, 2),
49     Book(3, 'Война', 450, 3),
```

```

50     Book(4, 'Ежик в тумане', 300, 3),
51     Book(5, 'Сказки', 200, 3),
52 ]
53
54 books_deps = [
55     BookShop(1, 1),
56     BookShop(2, 2),
57     BookShop(3, 3),
58     BookShop(3, 4),
59     BookShop(3, 5),
60
61     BookShop(11, 1),
62     BookShop(22, 2),
63     BookShop(33, 3),
64     BookShop(33, 4),
65     BookShop(33, 5),
66 ]
67
68 def first_task(one_to_many):
69     print('Задание A1')
70     res_11 = sorted(one_to_many, key=itemgetter(0))
71     print(res_11)
72     return res_11
73
74 def second_task(one_to_many):
75     print('\nЗадание A2')
76     res_12_unsorted = []
77     # Перебираем все магазины
78     for s in shops:
79         # Список книг магазина
80         s_books = list(filter(lambda i: i[2] == s.name, one_to_many))
81         # Если магазин не пустой
82         if len(s_books) > 0:
83             res_12_unsorted.append((s.name, len(s_books)))
84     # Сортировка по количеству книг
85     res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
86     print(res_12)
87     return res_12
88
89 def third_task(many_to_many):
90     print('\nЗадание A3')
91     res_13 = []
92     for i in many_to_many:
93         if i[0][-1:] == 'a':
94             res_13.append(i)
95     print(res_13)
96     return res_13
97
98 def main():
99     """Основная функция"""
100
101     # Соединение данных один-ко-многим
102     one_to_many = [(b.name, b.price, s.name)
103                     for s in shops

```

```

99         for s in shops
100         for b in books
101             if b.shop_id == s.id]
102
103     # Соединение данных многие-ко-многим
104     many_to_many_temp = [(s.name, bs.shop_id, bs.book_id)
105                          for s in shops
106                          for bs in books_deps
107                          if s.id == bs.shop_id]
108
109     many_to_many = [(b.name, b.price, shop_name)
110                    for shop_name, shop_id, book_id in many_to_many_temp
111                    for b in books if b.id == book_id]
112
113     first_task(one_to_many)
114     second_task(one_to_many)
115     third_task(many_to_many)
116
117
118
119     if __name__ == '__main__':
120         main()

```

testing.py

```

1  import unittest
2  from unittest import TestCase, main
3  from main import first_task, second_task, third_task
4
5  class Test(TestCase):
6      def test_task_1(self):
7          self.assertEqual(first_task([('Гарри Поттер', 250, 'Весёлая книжка'),
8                                     ('Незнайка', 300, 'Читатель'),
9                                     ('Война', 450, 'Книжный'),
10                                    ('Ежик в тумане', 300, 'Книжный'),
11                                    ('Сказки', 200, 'Книжный')]),
12                          [('Война', 450, 'Книжный'),
13                           ('Гарри Поттер', 250, 'Весёлая книжка'),
14                           ('Ежик в тумане', 300, 'Книжный'),
15                           ('Незнайка', 300, 'Читатель'),
16                           ('Сказки', 200, 'Книжный')])
17      def test_task_2(self):
18          self.assertEqual(second_task([('Гарри Поттер', 250, 'Весёлая книжка'),
19                                     ('Незнайка', 300, 'Читатель'),
20                                     ('Война', 450, 'Книжный'),
21                                     ('Ежик в тумане', 300, 'Книжный'),
22                                     ('Сказки', 200, 'Книжный')]),
23                          [('Книжный', 3),
24                           ('Весёлая книжка', 1),
25                           ('Читатель', 1)])
26      def test_task_3(self):
27          self.assertEqual(third_task([('Гарри Поттер', 250, 'Весёлая книжка'),
28                                     ('Незнайка', 300, 'Читатель'),
29                                     ('Война', 450, 'Книжный'),
30                                     ('Ежик в тумане', 300, 'Книжный'),
31                                     ('Сказки', 200, 'Книжный'),
32                                     ('Гарри Поттер', 250, 'Весёлая книжка (другой)'),
33                                     ('Незнайка', 300, 'Читатель (другой)'),
34                                     ('Война', 450, 'Книжный(другой)'),
35                                     ('Ежик в тумане', 300, 'Книжный(другой)'),
36                                     ('Сказки', 200, 'Книжный(другой)')]),
37                          [('Незнайка', 300, 'Читатель'),
38                           ('Война', 450, 'Книжный'),
39                           ('Незнайка', 300, 'Читатель (другой)'),
40                           ('Война', 450, 'Книжный(другой)')])
41
42     if __name__ == '__main__':
43         unittest.main()

```

Работа тестов:

```
...
-----
Ran 3 tests in 0.001s

OK
Задание A1
[('Война', 450, 'Книжный'), ('Гарри Поттер', 250, 'Весёлая книжка'), ('Ежик в тумане', 300, 'Книжный'), ('Незнайка', 300, 'Читатель'), ('Сказки', 200, 'Книжный')]

Задание A2
[('Книжный', 3), ('Весёлая книжка', 1), ('Читатель', 1)]

Задание A3
[('Незнайка', 300, 'Читатель'), ('Война', 450, 'Книжный'), ('Незнайка', 300, 'Читатель (другой)'), ('Война', 450, 'Книжный(другой)')]

Process finished with exit code 0
```