**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет**

**имени Н.Э. Баумана (национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»**

**Кафедра «Системы обработки информации и управления»**

**Курс «Базовые компоненты интернет-технологий»**

Отчёт по Рубежному Контролю №2

по курсу «Базовые компоненты и интернет-технологии»

Вариант 16

Выполнил:

студент группы ИУ5-33Б

Рыбин Владислав

Проверил:

к.т.н., доц., Ю. Е. Гапанюк

2022 г.

Полученное задание:

1) Провести рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.

2) Для текста программы рубежного контроля №1 создать модульные тесты с применением TDD - фреймворка (3 теста).

Текст программы:

from unittest import TestCase, main  
from rk1 import \*   
  
class order\_test(TestCase):  
 def test\_1(self):  
 expected\_result\_for\_C1 = [('Антикварная лавка', ['Курс Python', 'Анна Каренина']),  
 ('Аллегория', ['Евгений Онегин'])]  
 res\_C1 = []  
 for i in Shops:  
 book = list(filter(lambda a: a[2] == i.name, one\_to\_many))  
 if i.name[0] == 'А':  
 s\_name = [x for x, \_, \_ in book]  
 res\_C1.append((i.name, s\_name))  
 self.assertEqual(res\_C1, expected\_result\_for\_C1)  
  
 def test\_2(self):  
 expected\_result\_for\_C2 = [('Книжный лабиринт', 100),  
 ('Антикварная лавка', 179),  
 ('Читай-город', 254),  
 ('Аллегория', 660),  
 ('Достоевский', 700)]  
 res\_unsorted\_C2 = []  
 for s in Shops:  
 s\_book = list(filter(lambda i: i[2] == s.name, one\_to\_many))  
 if len(s\_book) > 0:  
 prices = [price for \_, price, \_ in s\_book]  
 prices\_min = min(prices)  
 res\_unsorted\_C2.append((s.name, prices\_min))  
 res\_C2 = sorted(res\_unsorted\_C2, key=itemgetter(1))  
 self.assertEqual(res\_C2, expected\_result\_for\_C2)  
  
 def test\_3(self):  
 expected\_result\_for\_C3 = [('Книжный лабиринт', 'Анжелика'),  
 ('Антикварная лавка', 'Анна Каренина'),  
 ('Достоевский', 'Всадник без головы'),  
 ('Аллегория', 'Евгений Онегин'),  
 ('Антикварная лавка', 'Курс Python'),  
 ('Читай-город', 'Левша'),  
 ('Читай-город', 'Повесть временных лет')]  
  
 # Перебираем все отделы  
 many\_to\_many.sort(key=lambda i: i[1])  
 self.assertEqual(many\_to\_many, expected\_result\_for\_C3)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

Результаты выполнения:

