

**Московский
государственный
технический университет
им. Н.Э. Баумана**

Факультет «Информатика и
системы управления»
Кафедра ИУ5 «Системы обработки
информации и управления»

Курс «Парадигмы и конструкции языков
программирования»

Отчет по Рубежному контролю №2

Выполнил:
студент группы ИУ5-34Б
Файзуллин Камиль
Подпись и дата:

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

Москва, 2023 г.

Задание:

Условия рубежного контроля №2 по курсу ПИК ЯП

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 из rk1 после рефакторинга:

```
from operator import itemgetter

class Detail:
    """Деталь"""
    def __init__(self, id, name, price, man_id):
        self.id = id
        self.name = name
        self.price = price
        self.man_id = man_id

class Manufacturer:
    """Производитель"""
    def __init__(self, id, title):
        self.id = id
        self.title = title

class DetMan:
    def __init__(self, man_id, det_id):
        self.man_id = man_id
        self.det_id = det_id

manufacturers = [
    Manufacturer(1, 'Дом в уюте'),
    Manufacturer(2, 'Строй в удовольствие'),
    Manufacturer(3, 'Домашний помощник'),
    Manufacturer(4, 'Леруа мерлен'),
]

details = [
    Detail(1, 'Подшипник', 600, 1),
    Detail(2, 'Колесо', 800, 1),
    Detail(3, 'Винт', 400, 2),
    Detail(4, 'Болт', 450, 3),
    Detail(5, 'Втулка', 700, 3),
]

det_man = [
    DetMan(1, 1),
    DetMan(1, 4),
    DetMan(2, 3),
    DetMan(3, 2),
    DetMan(3, 3),
    DetMan(3, 4),
    DetMan(4, 1),
    DetMan(4, 3),
    DetMan(5, 3),
    DetMan(5, 2),
]
```

```

]

def g1(o_to_m):
    titles = []
    for i in range(0, len(o_to_m)):
        if o_to_m[i][2][0] == "Д":
            titles.append(o_to_m[i][2])
    res_1 = {title: [(otm[0], otm[1]) for otm in o_to_m if otm[2] == title]
    for title in titles}
    return res_1

def g2(o_to_m):
    titles = []
    for i in range(0, len(o_to_m)):
        if o_to_m[i][2][0] == "Д":
            titles.append(o_to_m[i][2])
    mans = [m.title for m in manufacturers]
    res_2 = sorted([(title, max([otm[1] for otm in o_to_m if otm[2] ==
title])) for title in titles], key=itemgetter(1), reverse=True)
    return res_2

def g3(m_to_m):
    res_3 = sorted(m_to_m, key=itemgetter(2))
    return [res_3[i] for i in range(0, len(res_3))]

def main():
    o_to_m = [(d.name, d.price, m.title)
    for d in details
    for m in manufacturers
    if d.man_id == m.id]

    m_to_m = [(m.title, dm.man_id, dm.det_id)
    for m in manufacturers
    for dm in det_man
    if m.id == dm.man_id]

    m_to_m = [(d.name, d.price, man_title)
    for man_title, man_id, det_id in m_to_m
    for d in details if d.id == det_id]

    print('Задание Г1')
    print(g1(o_to_m))

    print('\nЗадание Г2')
    print(g2(o_to_m))

    print('\nЗадание Г3')
    print(g3(m_to_m))

if __name__ == '__main__':
    main()

```

Файл rk2:

```

import unittest
from rk1 import *
class test_prog(unittest.TestCase):

```

```

manufacturers = [
    Manufacturer(1, 'Дом в уютe'),
    Manufacturer(2, 'Строй в удовольствие'),
    Manufacturer(3, 'Домашний помощник'),
    Manufacturer(4, 'Леруа мерлен'),
]

details = [
    Detail(1, 'Подшипник', 600, 1),
    Detail(2, 'Колесо', 800, 1),
    Detail(3, 'Винт', 400, 2),
    Detail(4, 'Болт', 450, 3),
    Detail(5, 'Втулка', 700, 3),
]

det_man = [
    DetMan(1, 1),
    DetMan(1, 4),
    DetMan(2, 3),
    DetMan(3, 2),
    DetMan(3, 3),
    DetMan(3, 4),
    DetMan(4, 1),
    DetMan(4, 3),
    DetMan(5, 3),
    DetMan(5, 2),
]

def test_g1(self):
    o_to_m = [(d.name, d.price, m.title) for d in details for m in
manufacturers if d.man_id == m.id]
    self.assertEqual(g1(o_to_m), {'Дом в уютe': [('Подшипник', 600),
('Колесо', 800)], 'Домашний помощник': [('Болт', 450), ('Втулка', 700)]})

def test_g2(self):
    o_to_m = [(d.name, d.price, m.title) for d in details for m in
manufacturers if d.man_id == m.id]
    self.assertEqual(g2(o_to_m), [('Дом в уютe', 800), ('Дом в уютe',
800), ('Домашний помощник', 700), ('Домашний помощник', 700)])

def test_g3(self):
    m_to_m = [(m.title, dm.man_id, dm.det_id) for m in manufacturers for
dm in det_man if m.id == dm.man_id]
    m_to_m = [(d.name, d.price, man_title) for man_title, man_id,
det_id in m_to_m for d in details if d.id == det_id]
    self.assertEqual(g3(m_to_m), [('Подшипник', 600, 'Дом в уютe'),
('Болт', 450, 'Дом в уютe'), ('Колесо', 800, 'Домашний помощник'), ('Винт',
400, 'Домашний помощник'), ('Болт', 450, 'Домашний помощник'), ('Подшипник',
600, 'Леруа мерлен'), ('Винт', 400, 'Леруа мерлен'), ('Винт', 400, 'Строй в
удовольствие')])

if __name__ == '__main__':
    main()

```

Результат:

```
Ran 3 tests in 0.003s
```

```
OK
```

```
Launching unittests with arguments python -m unittest rk2.test_prog in C:\Users\Home\PycharmProjects\sem3 lab Python
```

```
Process finished with exit code 0
```

✖ Tests failed: 1, passed: 2 of 3 tests – 300 ms

```
Ran 3 tests in 0.301s
```

```
FAILED (failures=1)
```

```
{'Дом в уютe': [('Подшипник', 600), ('Колесо', 800)],  
'Домашний помощник': [('Болт', 450), ('Втулка', 700)]} != {'Дом в уютe': [('Подшипник', 600), ('Колесо', 800)],  
'Домашний помощник': [('Болт', 450), ('Втулка', 700)]}
```