

**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»

Рубежный контроль №2

Вариант Г №19

Выполнил:

Студент ИУ5-34Б

Флоринский В. А.

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).

Текст программ.

main.py:

```
from prettytable import PrettyTable
from operator import itemgetter

class Detail:
    def __init__(self, id, name, price, sup_id):
        self.id = id
        self.name = name
        self.price = price
        self.sup_id = sup_id

class Suppliers:
    def __init__(self, id, name):
        self.id = id
        self.name = name

class DetSup:
    def __init__(self, sup_id, det_id):
        self.sup_id = sup_id
        self.det_id = det_id

def a1_solution(one_to_many):
    arr = sorted(one_to_many, key=itemgetter(2))
    ans = []
    for el in arr:
        if el[2][0] == 'A':
            ans.append(el)
    if len(ans) != 0 : return (ans)
    else : return "Отделов на А нет!"

def a2_solution(one_to_many):
    arr0 = []
    for d in suppliers:
        d_dets = list(filter(lambda i: i[2]==d.name, one_to_many))
        if len(d_dets) > 0:
            d_prices = [price for _,price,_ in d_dets]
            d_sals_sum = sum(d_prices)
            arr0.append((d.name, d_sals_sum))
    res = sorted(arr0, key=itemgetter(1), reverse=True)
    return res
```

```

def a3_solution(many_to_many):
    arr = {}
    for d in suppliers:
        d_emps = list(filter(lambda i: i[2]==d.name, many_to_many))
        d_emps_names = [x for x,_,_ in d_emps]
        arr[d.name] = d_emps_names
    arr = dict(sorted(arr.items()))
    return arr

suppliers = [
    Suppliers(1, "Звезда-Техника"),
    Suppliers(2, "Сибирские Детали"),
    Suppliers(3, "Волжский Агрегат"),
    Suppliers(4, "АгроПром Комбинат"),
]

details = [
    Detail(1, "Винт", 1500, 1),
    Detail(2, "Гайка", 1700, 2),
    Detail(3, "Гайка", 1900, 3),
    Detail(4, "Шайба", 2000, 4),
    Detail(5, "Шплинт", 2100, 4),
]

detsup = [
    DetSup(1, 1),
    DetSup(1, 2),
    DetSup(2, 2),
    DetSup(2, 3),
    DetSup(3, 1),
    DetSup(3, 4),
    DetSup(4, 2),
    DetSup(4, 5),
]

def main():
    one_to_many = [(p.name, p.price, d.name)
                   for d in suppliers
                   for p in details
                   if p.sup_id == d.id]

    many_to_many_temp = [(d.name, ed.sup_id, ed.det_id)
                         for d in suppliers
                         for ed in detsup
                         if d.id == ed.sup_id]

    many_to_many = [(e.name, e.price, sup_name)
                    for sup_name, sup_id, det_id in many_to_many_temp
                    for e in details
                    if e.id == det_id]

```

```

print('Задание A1')
print(a1_solution(one_to_many))

print('\nЗадание A2')
print(a2_solution(one_to_many))

print('\nЗадание A3')
print(a3_solution(many_to_many))

main()

```

test.py:

```

import unittest
from main import *

class Test_Program(unittest.TestCase):

    suppliers = [
        Suppliers(1, "Звезда-Техника"),
        Suppliers(2, "Сибирские Детали"),
        Suppliers(3, "Волжский Агрегат"),
        Suppliers(4, "АгроПром Комбинат"),
    ]

    details = [
        Detail(1, "Винт", 1500, 1),
        Detail(2, "Гайка", 1700, 2),
        Detail(3, "Гайка", 1900, 3),
        Detail(4, "Шайба", 2000, 4),
        Detail(5, "Шплинт", 2100, 4),
    ]

    detsup = [
        DetSup(1, 1),
        DetSup(1, 2),
        DetSup(2, 2),
        DetSup(2, 3),
        DetSup(3, 1),
        DetSup(3, 4),
        DetSup(4, 2),
        DetSup(4, 5),
    ]

    def test_a1(self):
        one_to_many = [(p.name, p.price, d.name)
                        for d in suppliers
                        for p in details
                        if p.sup_id == d.id]

```

```

self.assertEqual(a1_solution(one_to_many),
                 [('Шайба', 2000, 'АгроПром Комбинат'),
                  ('Шплинт', 2100, 'АгроПром Комбинат')])

def test_a2(self):
    one_to_many = [(p.name, p.price, d.name)
                    for d in suppliers
                    for p in details
                    if p.sup_id == d.id]

    self.assertEqual(a2_solution(one_to_many),
                     [('АгроПром Комбинат', 4100),
                      ('Волжский Агрекат', 1900),
                      ('Сибирские Детали', 1700),
                      ('Звезда-Техника', 1500)])

def test_a3(self):
    many_to_many_temp = [(d.name, ed.sup_id, ed.det_id)
                          for d in suppliers
                          for ed in detsup
                          if d.id == ed.sup_id]

    many_to_many = [(e.name, e.price, sup_name)
                     for sup_name, sup_id, det_id in many_to_many_temp
                     for e in details
                     if e.id == det_id]

    self.assertEqual(a3_solution(many_to_many),
                     {'АгроПром Комбинат': ['Гайка', 'Шплинт'],
                      'Волжский Агрекат': ['Винт', 'Шайба'],
                      'Звезда-Техника': ['Винт', 'Гайка'],
                      'Сибирские Детали': ['Гайка', 'Гайка']})

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

```

Результаты Тестов.

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

```
OK
```

Если при проверке функции `a3_solution` намеренно сделать ошибку в тестах, то в результатах теста увидим ошибку:

```
{ 'АгроПром Комбинат': ['Гайка', 'Шплинт'],  
- 'Волжский Агрегат': ['Винт', 'Шайба'],  
?   ^  
  
+ 'Валжский Агрегат': ['Винт', 'Шайба'],  
?   ^  
  
  'Звезда-Техника': ['Винт', 'Гайка'],  
  'Сибирские Детали': ['Гайка', 'Гайка']}
```

Ran 3 tests in 0.002s

FAILED (failures=1)