Студент: Секретов К. А.
Группа: ИУ5-32Б

РК2 по БКИТ

Вариант запросов: Б
Вариант предметной области: 20

Задания:

1) Проведите рефакторинг текста программы рубежного контроля №1 такимобразом, 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

rk.py

```
from audioop import reverse
from operator import itemgetter
class Det:
    def __init__(self, id, name, snumber):
        self.id = id
        self.name = name
        self.snumber = snumber
class Sup:
    def __init__(self, id, name, totalShopers, det_id):
        self.id = id
        self.name = name
        self.totalShopers = totalShopers
        self.det_id = det_id
class DetToComp:
    def __init__(self, Det_id, Comp_id):
        self.det_id = Det_id
        self.Comp_id = Comp_id
Details = [
    Det(0, "Гвоздь", "А3"),
    Det(1, "Гайка", "ГГ"),
    Det(2, "Шуруп", "Ш2.2"),
    Det(3, "Шайба", "Ш3.1"),
    Det(4, "Болт", "Б5"),
    Det(5, "Шестеренка", "Ш5.1"),
    Det(6, "Заглушка", "З1"),
]
```

```
Suppliers = [
    Sup(0, "СТРОЙМАГ", 145000,1),
    Sup(1, "СТРОИТЕЛЬНЫЙ МИР", 980000,2),
    Sup(2, "МИРОВАЯ СТРОЙКА",111000,3),
    Sup(3, "КВАДРАТ К КУБЕ", 354000,4),
    Sup(4, "ХОЗМАГ", 4850000,4),
    Sup(5, "Магнит", 3457000,2),
    Sup(6, "Дикси", 2240000,6),
]
DetailsNCompanies = [
DetToComp(0,0),
DetToComp(0,1),
DetToComp(0,2),
DetToComp(1,3),
DetToComp(2,3),
DetToComp(2,4),
DetToComp(2,5),
DetToComp(3,6),
DetToComp(3,1),
DetToComp(4,0),
DetToComp(4,1),
DetToComp(4,4),
DetToComp(5,4),
DetToComp(5,6),
DetToComp(6,0),
DetToComp(6,3),
DetToComp(6,6),
DetToComp(6,2),
]
def count(det_id):
    c = 0
    for i in Suppliers:
        if i.det_id == det_id:
            c+=1
    return c


def task1(one_to_many):
    B1 = sorted(one_to_many,key = itemgetter(1),reverse=True)
    print('Б1:')
    for row in B1:
        print(row[:-1])
```

```
    return B1

def task2(one_to_many):
    B2 = []
    for n in Details:
        l_Supplier = list(filter(lambda i: i[4]==n.id, one_to_many))
        if len(l_Supplier) > 0 : B2.append((n.name, len(l_Supplier)))
    B22 = sorted(B2,key = lambda i: i[1], reverse = True)
    print('Б2')
    for row in B22:
        print(row)
    return B22

def task3(many_to_many):
    B3 = {}
    for n in Details:
        #if n.name[-1] == "a":
        if True==True:
            lsup = list(filter(lambda i: i[1] == n.id, many_to_many))
            l_Supplier_names = [x for _, _, x in lsup]
            B3[n.name] = l_Supplier_names
    print('Б3')
    print(B3)
    return B3



def main():
    one_to_many = [(i.name, i.totalShopers, n.name, n.snumber, n.id)
        for n in Details
        for i in Suppliers
        if i.det_id==n.id]
    many_to_many_temp = [(i.name, li.Comp_id, li.det_id)
        for i in Suppliers
        for li in DetailsNCompanies
        if li.Comp_id == i.id]
    many_to_many = [(n.name, det_id, supp_name)
        for supp_name, Comp_id, det_id in many_to_many_temp
        for n in Details if n.id == det_id]
    task1(one_to_many)
    task2(one_to_many)
    task3(many_to_many)
```

```
main()
```

test.py

```
from unittest import TestCase, main
from rk import task1, task2, task3

class testrk2(TestCase):
   def test_task1(self):
      self.assertEqual(task1(
         [('МИРОВАЯ СТРОЙКА', 111000, 'Шайба', 'Ш3.1', 3),
          ('СТРОИТЕЛЬНЫЙ МИР', 980000, 'Шуруп', 'Ш2.2', 2),
          ('КВАДРАТ К КУБЕ', 354000, 'Болт', 'Б5', 4),
          ('ХОЗМАГ', 4850000, 'Болт', 'Б5', 4),
          ('СТРОЙМАГ', 145000, 'Гайка', 'ГГ', 1),
          ('Магнит', 3457000, 'Шуруп', 'Ш2.2', 2),
          ('Дикси', 2240000, 'Заглушка', 'З1', 6)]),

         [('ХОЗМАГ', 4850000, 'Болт', 'Б5', 4),
          ('Магнит', 3457000, 'Шуруп', 'Ш2.2', 2),
          ('Дикси', 2240000, 'Заглушка', 'З1', 6),
          ('СТРОИТЕЛЬНЫЙ МИР', 980000, 'Шуруп', 'Ш2.2', 2),
          ('КВАДРАТ К КУБЕ', 354000, 'Болт', 'Б5', 4),
          ('СТРОЙМАГ', 145000, 'Гайка', 'ГГ', 1),
          ('МИРОВАЯ СТРОЙКА', 111000, 'Шайба', 'Ш3.1', 3)])

   def test_task2(self):
      self.assertEqual(task2(
         [('МИРОВАЯ СТРОЙКА', 111000, 'Шайба', 'Ш3.1', 3),
          ('СТРОИТЕЛЬНЫЙ МИР', 980000, 'Шуруп', 'Ш2.2', 2),
          ('КВАДРАТ К КУБЕ', 354000, 'Болт', 'Б5', 4),
          ('ХОЗМАГ', 4850000, 'Болт', 'Б5', 4),
          ('СТРОЙМАГ', 145000, 'Гайка', 'ГГ', 1),
          ('Магнит', 3457000, 'Шуруп', 'Ш2.2', 2),
          ('Дикси', 2240000, 'Заглушка', 'З1', 6)]),

         [('Шуруп', 2),
('Болт', 2),
('Гайка', 1),
('Шайба', 1),
('Заглушка', 1)])
```

```
    def test_task3(self):
        self.assertEqual(task3(
            [('Гвоздь', 0, 'СТРОЙМАГ'),
             ('Болт', 4, 'СТРОЙМАГ'),
             ('Заглушка', 6, 'СТРОЙМАГ'),
             ('Гвоздь', 0, 'СТРОИТЕЛЬНЫЙ МИР'),
             ('Шайба', 3, 'СТРОИТЕЛЬНЫЙ МИР'),
             ('Болт', 4, 'СТРОИТЕЛЬНЫЙ МИР'),
             ('Гвоздь', 0, 'МИРОВАЯ СТРОЙКА'),
             ('Заглушка', 6, 'МИРОВАЯ СТРОЙКА'),
             ('Гайка', 1, 'КВАДРАТ К КУБЕ'),
             ('Шуруп', 2, 'КВАДРАТ К КУБЕ'),
             ('Заглушка', 6, 'КВАДРАТ К КУБЕ'),
             ('Шуруп', 2, 'ХОЗМАГ'),
             ('Болт', 4, 'ХОЗМАГ'),
             ('Шестеренка', 5, 'ХОЗМАГ'),
             ('Шуруп', 2, 'Магнит'),
             ('Шайба', 3, 'Дикси'),
             ('Шестеренка', 5, 'Дикси'),
             ('Заглушка', 6, 'Дикси')]),


            {'Гвоздь': ['СТРОЙМАГ', 'СТРОИТЕЛЬНЫЙ МИР', 'МИРОВАЯ СТРОЙКА'],
             'Гайка': ['КВАДРАТ К КУБЕ'],
             'Шуруп': ['КВАДРАТ К КУБЕ', 'ХОЗМАГ', 'Магнит'],
             'Шайба': ['СТРОИТЕЛЬНЫЙ МИР', 'Дикси'],
             'Болт': ['СТРОЙМАГ', 'СТРОИТЕЛЬНЫЙ МИР', 'ХОЗМАГ'],
             'Шестеренка': ['ХОЗМАГ', 'Дикси'],
             'Заглушка': ['СТРОЙМАГ', 'МИРОВАЯ СТРОЙКА', 'КВАДРАТ К КУБЕ',
'Дикси']})

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

Результаты выполнения:

```
Б1:
('ХОЗМАГ', 4850000, 'Болт', 'Б5')
('Магнит', 3457000, 'Шуруп', 'Ш2.2')
('Дикси', 2240000, 'Заглушка', 'З1')
('СТРОИТЕЛЬНЫЙ МИР', 980000, 'Шуруп', 'Ш2.2')
('КВАДРАТ К КУБЕ', 354000, 'Болт', 'Б5')
('СТРОЙМАГ', 145000, 'Гайка', 'ГГ')
('МИРОВАЯ СТРОЙКА', 111000, 'Шайба', 'Ш3.1')
.Б2
('Шуруп', 2)
('Болт', 2)
('Гайка', 1)
('Шайба', 1)
('Заглушка', 1)
.Б3
{'Гвоздь': ['СТРОЙМАГ', 'СТРОИТЕЛЬНЫЙ МИР', 'МИРОВАЯ СТРОЙКА'], 'Гайка': ['КВАДРАТ К КУБЕ'], 'Шуруп': ['КВАДРАТ К КУБЕ',
'ХОЗМАГ', 'Магнит'], 'Шайба': ['СТРОИТЕЛЬНЫЙ МИР', 'Дикси'], 'Болт': ['СТРОЙМАГ', 'СТРОИТЕЛЬНЫЙ МИР', 'ХОЗМАГ'], 'Шестер
енка': ['ХОЗМАГ', 'Дикси'], 'Заглушка': ['СТРОЙМАГ', 'МИРОВАЯ СТРОЙКА', 'КВАДРАТ К КУБЕ', 'Дикси']}
.
----------------------------------------------------------------------
Ran 3 tests in 0.203s

OK
>>>
```