

РК №2 по БКИТ

Задание

№1: Провести рефакторинг текста программы РК№1 таким образом, чтобы он был пригоден для модульного тестирования

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

Задание для РК№1

№1: "Компьютер" и "Дисплейный класс" связаны соотношениями один-ко-многим. Отсортировать компьютеры по популярности и вывести компьютеры и дисплеи.

№2: "Компьютер" и "Дисплейный класс" связаны соотношениями один-ко-многим. Отсортировать компьютеры по количеству подключенных к ним дисплеев.

№3: "Компьютер" и "Дисплейный класс" связаны соотношениями многие-ко-многим. Вывести все компьютеры, названия которых начинаются на "А" и названия мониторов, подключенных к ним.

Изменённый текст программы

main.py

```
#используется для сортировки
from audioop import reverse
from operator import itemgetter
class Comp:
    def __init__(self, id, name, year):
        self.id = id
        self.name = name
        self.year = year
class Disps:
    def __init__(self, id, name, diagonal, con_int):
        self.id = id
        self.name = name
        self.diagonal = diagonal
        self.con_int = con_int
class CompDisps:
    def __init__(self, comp_id, disp_id):
```

```

        self.comp_id = comp_id
        self.disp_id = disp_id
computers = [
    Comp(0, "HP", "2019"),
    Comp(1, "Apple", "2022"),
    Comp(2, "Asus", "2013"),
    Comp(3, "Lenovo", "2020"),
    Comp(4, "MSI", "2018"),
    Comp(5, "Acer", "2021"),
    Comp(6, "Microsoft", "2022"),
    Comp(7, "Razer", "2018"),
    Comp(8, "Xiaomi", "2016"),
]
display_class = [
    Disps(0, "Samsung", 27, "DP"),
    Disps(1, "Asus", 24, "HDMI"),
    Disps(2, "Acer", 21, "HDMI"),
    Disps(3, "Benq", 34, "DP"),
    Disps(4, "MSI", 21, "DP"),
    Disps(5, "AOC", 24, "DVI"),
    Disps(6, "Gigabyte", 13, "DVI"),
]
comp_dclass = [
    CompDisps(0,0),
    CompDisps(0,1),
    CompDisps(0,3),
    CompDisps(1,5),
    CompDisps(2,1),
    CompDisps(2,5),
    CompDisps(2,6),
    CompDisps(3,1),
    CompDisps(3,2),
    CompDisps(4,4),
    CompDisps(5,0),
    CompDisps(5,6),
    CompDisps(6,0),
    CompDisps(6,6),
    CompDisps(7,1),
    CompDisps(8,0),
    CompDisps(8,1),
]

def counter(disp_id):
    count = 0
    for i in display_class:
        if i.disp_id == disp_id:
            count+=1
    return count

def task1(one_to_many):
    B1 = sorted(one_to_many, key = itemgetter(1), reverse=True)

```

```

print('-----')
print('Задание Б1')
print('-----')
for row in B1:
    print(row[:-1])
return B1

def task2(one_to_many):
    B2 = []
    for l in computers:
        l_ids = list(filter(lambda i: i[4]==l.id, one_to_many))
        if len(l_ids) > 0 : B2.append((l.name, len(l_ids)))
        newB2 = sorted(B2, key = lambda i: i[1], reverse = True)
    print('-----')
    print('Задание Б2')
    print('-----')
    for row in newB2:
        print(row)
    return B2

def task3(many_to_many):
    B3 = {}

    for l in computers:
        if l.name[0] == "A":
            lst_IDE = list(filter(lambda i: i[1] == l.id,
                                   many_to_many))
            l_ids_names = [x for _, x in lst_IDE]
            B3[l.name] = l_ids_names

    print('-----')
    print('Задание Б3')
    print('-----')
    print(B3)
    return B3

def main():
    #Соединение данных один-ко-многим
    one_to_many = [(i.name, i.diagonal, l.name, l.year, l.id)
                    for l in computers
                    for i in display_class
                    if i.id==l.id]

    #Задание Б2

    many_to_many_temp = [(i.name, li.disp_id, li.comp_id)
                          for i in display_class
                          for li in comp_dclass
                          if li.disp_id == i.id]

    many_to_many = [(l.name, comp_id, IDE_name)

```

```

        for IDE_name, disp_id, comp_id in many_to_many_temp
        for l in computers if l.id == comp_id]

for i in many_to_many:
    print(str(i) + ",")

task1(one_to_many)
task2(one_to_many)
task3(many_to_many)

main()

```

test.py

```

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

class testrk2(TestCase):
    def test_task1(self):
        self.assertEqual(task1(
            [('Benq', 34, 'Lenovo', '2020'),
             ('Samsung', 27, 'HP', '2019'),
             ('Asus', 24, 'Apple', '2022'),
             ('AOC', 24, 'Acer', '2021'),
             ('Acer', 21, 'Asus', '2013'),
             ('MSI', 21, 'MSI', '2018'),
             ('Gigabyte', 13, 'Microsoft', '2022')]),
            [('Benq', 34, 'Lenovo', '2020'),
             ('Samsung', 27, 'HP', '2019'),
             ('Asus', 24, 'Apple', '2022'),
             ('AOC', 24, 'Acer', '2021'),
             ('Acer', 21, 'Asus', '2013'),
             ('MSI', 21, 'MSI', '2018'),
             ('Gigabyte', 13, 'Microsoft', '2022')])

    def test_task2(self):
        self.assertEqual(task2(
            [('Benq', 34, 'Lenovo', '2020'),
             ('Samsung', 27, 'HP', '2019'),
             ('Asus', 24, 'Apple', '2022'),
             ('AOC', 24, 'Acer', '2021'),
             ('Acer', 21, 'Asus', '2013'),
             ('MSI', 21, 'MSI', '2018'),
             ('Gigabyte', 13, 'Microsoft', '2022')]),
            [('Acer', 2),
             ('HP', 1),
             ('Apple', 1),
             ('Asus', 1),

```

```

        ('Lenovo', 1),
        ('MSI', 1),
        ('Microsoft', 1)])

def test_task3(self):
    self.assertEqual(task3(
        [('HP', 0, 'Samsung'),
         ('Acer', 5, 'Samsung'),
         ('Microsoft', 6, 'Samsung'),
         ('Xiaomi', 8, 'Samsung'),
         ('HP', 0, 'Asus'),
         ('Asus', 2, 'Asus'),
         ('Lenovo', 3, 'Asus'),
         ('Razer', 7, 'Asus'),
         ('Xiaomi', 8, 'Asus'),
         ('Lenovo', 3, 'Acer'),
         ('HP', 0, 'Benq'),
         ('MSI', 4, 'MSI'),
         ('Apple', 1, 'AOC'),
         ('Asus', 2, 'AOC'),
         ('Asus', 2, 'Gigabyte'),
         ('Acer', 5, 'Gigabyte'),
         ('Microsoft', 6, 'Gigabyte')]),

        {'Apple': ['AOC'],
         'Asus': ['Asus', 'AOC', 'Gigabyte'],
         'Acer': ['Samsung', 'Gigabyte']})

if __name__ == "__main__":
    main()

```

Результаты прохождения тестов

```

c:/Users/vaski/OneDrive/Рабочий стол/Прога 2 курс/RK2 _my/RK2/test.py"
-----
Задание Б1
-----
('Benq', 34, 'Lenovo', '2020')
('Samsung', 27, 'HP', '2019')
('Asus', 24, 'Apple', '2022')
('AOC', 24, 'Acer', '2021')
('Acer', 21, 'Asus', '2013')
('MSI', 21, 'MSI', '2018')
('Gigabyte', 13, 'Microsoft', '2022')
-----
Задание Б2
-----
('HP', 1)
('Apple', 1)

```

```
('Asus', 1)
('Lenovo', 1)
('MSI', 1)
('Acer', 1)
('Microsoft', 1)
```

Задание Б3

```
{'Apple': ['AOC'], 'Asus': ['Asus', 'AOC', 'Gigabyte'], 'Acer': ['Samsung',
'Gigabyte']}
```

Ran 3 tests in 0.002s

OK