Марянян Артур ИУ5-34Б

**Отчет РК-2 по дисциплине**

**Парадигмы и конструкторы языков программирования**

Задание

1. Проведите рефакторинг текста программы таким образом, чтобы он был пригоден для тестирования

2. Для текста программы создайте модульное тестирование с применением TDD-фреймворка from operator import itemgetter

Текст программы main.py

from operator import itemgetter

class Microprocessor:

    def \_\_init\_\_(*self*, *id*, *count*, *price*, *comp\_id*):

*self*.id = *id*

*self*.count = *count*

*self*.price = *price*

*self*.comp\_id = *comp\_id*

class Computer:

    def \_\_init\_\_(*self*, *id*, *name*):

*self*.id = *id*

*self*.name = *name*

class MicrComp:

    def \_\_init\_\_(*self*, *micr\_id*, *comp\_id*):

*self*.micr\_id = *micr\_id*

*self*.comp\_id = *comp\_id*

def get\_one\_to\_many(*microprocessors*, *computers*):

    return [

        (m.count, m.price, c.name)

        for m in *microprocessors*

        for c in *computers*

        if m.comp\_id == c.id

    ]

def get\_many\_to\_many(*microprocessors*, *computers*, *micr\_comp*):

    many\_to\_many\_temp = [

        (c.name, mc.micr\_id, mc.comp\_id)

        for c in *computers*

        for mc in *micr\_comp*

        if c.id == mc.comp\_id

    ]

    return [

        (m.count, m.price, c.name)

        for mc in *micr\_comp*

        for m in *microprocessors* if m.id == mc.micr\_id

        for c in *computers* if c.id == mc.comp\_id

    ]

def task\_g1(*computers*, *microprocessors*):

    result = {}

    for c in *computers*:

        if int(c.name[-1]) < 4:

            m\_c = [

                (m.count, m.price) for m in *microprocessors* if m.comp\_id == c.id

            ]

            result[c.name] = m\_c

    return result

def task\_g2(*computers*, *one\_to\_many*):

    result = []

    for c in *computers*:

        c\_micrs = list(filter(lambda *i*: *i*[2] == c.name, *one\_to\_many*))

        if c\_micrs:

            s\_price = [price for \_, price, \_ in c\_micrs]

            s\_max = max(s\_price)

            result.append((c.name, s\_max))

    return sorted(result, *key*=itemgetter(1), *reverse*=True)

def task\_g3(*many\_to\_many*):

    return sorted(*many\_to\_many*, *key*=itemgetter(2))

# Пример использования

def main():

    computers = [

        Computer(1, 'Компьютер 1'),

        Computer(2, 'Компьютер 2'),

        Computer(3, 'Компьютер 3'),

        Computer(4, 'Компьютер 4'),

        Computer(5, 'Компьютер 5'),

        Computer(6, 'Компьютер 6'),

    ]

    microprocessors = [

        Microprocessor(1, 1001, 12000, 2),

        Microprocessor(2, 270011, 12442, 3),

        Microprocessor(3, 323312, 147977, 1),

        Microprocessor(4, 664623, 2356, 3),

        Microprocessor(5, 374223, 2467, 4),

        Microprocessor(6, 12654, 2357, 5),

    ]

    micr\_comp = [

        MicrComp(1, 1),

        MicrComp(2, 2),

        MicrComp(3, 3),

        MicrComp(3, 4),

        MicrComp(4, 4),

        MicrComp(5, 6),

        MicrComp(5, 2),

        MicrComp(1, 2),

    ]

    one\_to\_many = get\_one\_to\_many(microprocessors, computers)

    many\_to\_many = get\_many\_to\_many(microprocessors, computers, micr\_comp)

    print('Задание Г1:', task\_g1(computers, microprocessors))

    print('Задание Г2:', task\_g2(computers, one\_to\_many))

    print('Задание Г3:', task\_g3(many\_to\_many))

if \_\_name\_\_ == '\_\_main\_\_':

    main()

Текст программы tests.py

import unittest

from main import Computer, Microprocessor, MicrComp, get\_one\_to\_many, get\_many\_to\_many, task\_g1, task\_g2, task\_g3

class TestMicroprocessorFunctions(*unittest*.TestCase):

    def setUp(*self*):

*self*.computers = [

            Computer(1, 'Компьютер 1'),

            Computer(2, 'Компьютер 2'),

            Computer(3, 'Компьютер 3'),

            Computer(4, 'Компьютер 4'),

            Computer(5, 'Компьютер 5'),

            Computer(6, 'Компьютер 6'),

        ]

*self*.microprocessors = [

            Microprocessor(1, 1001, 12000, 2),

            Microprocessor(2, 270011, 12442, 3),

            Microprocessor(3, 323312, 147977, 1),

            Microprocessor(4, 664623, 2356, 3),

            Microprocessor(5, 374223, 2467, 4),

            Microprocessor(6, 12654, 2357, 5),

        ]

*self*.micr\_comp = [

            MicrComp(1, 1),

            MicrComp(2, 2),

            MicrComp(3, 3),

            MicrComp(3, 4),

            MicrComp(4, 4),

            MicrComp(5, 6),

            MicrComp(5, 2),

            MicrComp(1, 2),

        ]

    def test\_get\_one\_to\_many(*self*):

        result = get\_one\_to\_many(*self*.microprocessors, *self*.computers)

*self*.assertTrue(len(result) > 0)

*self*.assertIn((1001, 12000, 'Компьютер 2'), result)

    def test\_get\_many\_to\_many(*self*):

        result = get\_many\_to\_many(*self*.microprocessors, *self*.computers, *self*.micr\_comp)

*self*.assertTrue(len(result) > 0)

*self*.assertIn((1001, 12000, 'Компьютер 1'), result)

    def test\_task\_g1(*self*):

        result = task\_g1(*self*.computers, *self*.microprocessors)

*self*.assertIn('Компьютер 1', result)

    def test\_task\_g2(*self*):

        one\_to\_many = get\_one\_to\_many(*self*.microprocessors, *self*.computers)

        result = task\_g2(*self*.computers, one\_to\_many)

*self*.assertTrue(result[0][1] > result[-1][1])

    def test\_task\_g3(*self*):

        many\_to\_many = get\_many\_to\_many(*self*.microprocessors, *self*.computers, *self*.micr\_comp)

        result = task\_g3(many\_to\_many)

*self*.assertTrue(len(result) > 0)

if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()

Вывод программы

Задание Г1: {'Компьютер 1': [(323312, 147977)], 'Компьютер 2': [(1001, 12000)], 'Компьютер 3': [(270011, 12442), (664623, 2356)]}

Задание Г2: [('Компьютер 1', 147977), ('Компьютер 3', 12442), ('Компьютер 2', 12000), ('Компьютер 4', 2467), ('Компьютер 5', 2357)]

Задание Г3: [(1001, 12000, 'Компьютер 1'), (270011, 12442, 'Компьютер 2'), (374223, 2467, 'Компьютер 2'), (1001, 12000, 'Компьютер 2'), (323312, 147977, 'Компьютер 3'), (323312, 147977, 'Компьютер 4'), (664623, 2356, 'Компьютер 4'), (374223, 2467, 'Компьютер 6')]

....

----------------------------------------------------------------------

Ran 5 tests in 0.000s

OK