Лаптев Григорий ИУ5-35Б

Рубежный контроль №2 по курсу БКИТ

Вариант 11Д

Код из РК1, преобразованный для модульного тестирования:

```python
# tasks.py
from operator import itemgetter

class Programm:
    """Программа"""
    def __init__(self, id, name, size, com_id):
        self.id = id
        self.name = name
        self.size = size
        self.com_id = com_id

class Computer:
    """Компьютер"""
    def __init__(self, id, name, dev, display_size, storage, exp_storage):
        self.id = id
        self.name = name
        self.dev = dev
        self.display_size = display_size
        self.storage = storage
        self.exp_storage = exp_storage

class PrCom:
    """
    'Программы компьютера' для реализации
    связи многие-ко-многим
    """
    def __init__(self, Pr_id, Com_id):
        self.Pr_id = Pr_id
        self.Com_id = Com_id

computers = [
    Computer(1, 'Zenbook', 'Asus', 15.6, 512, 'MicroSD'),
    Computer(2, 'MacBook', 'Apple', 13.3, 256, 'SSD'),
    Computer(3, 'Inspiron', 'Dell', 15.6, 1024, 'HDD'),
    Computer(4, 'Latitude', 'Dell', 15.6, 1024, 'SSD'),
    Computer(5, 'ThinkPad', 'Lenovo', 15.6, 1024, 'SSD'),
    Computer(6, 'ProBook', 'HP', 15.6, 1024, 'SSD'),
]

programms = [
    Programm(1, 'Word', 110, 1),
    Programm(2, 'Excel', 120, 2),
    Programm(3, 'Access', 140, 3),
    Programm(4, 'MultiSim', 50, 3),
    Programm(5, 'Zoom', 100, 3),
    Programm(6, 'Chrome', 90, 3)
]

PrComs = [
```

```
    PrCom(1, 1),
    PrCom(1, 3),
    PrCom(1, 2),
    PrCom(2, 1),
    PrCom(2, 5),
    PrCom(3, 6),
    PrCom(4, 4),
    PrCom(4, 6),
    PrCom(5, 6),
    PrCom(5, 2),
    PrCom(6, 1),
    PrCom(6, 6),
    PrCom(6, 3),

]

one_to_many = [(c.name, b.name, b.size)
                   for b in programms
                   for c in computers
                   if b.com_id == c.id]
many_to_many_curr = [(c.name, pc.Com_id, pc.Pr_id)
                     for c in computers
                     for pc in PrComs
                     if c.id == pc.Com_id]
many_to_many = [(com_name, b.name)
                for com_name, com_id, prog_id in many_to_many_curr
                for b in programms
                if b.id == prog_id]
def task_1():
    return [i for i in one_to_many if i[1][- 1]== 'm']

def task_2():
    s = 0
    count = 0
    avg = []
    for c in computers:
        for b in programms:
            if b.com_id == c.id:
                count = count + 1
                s += b.size
                avg.append((c.name, '%.2f' % (s / count)))
    result2 = sorted(avg, key=itemgetter(1))
    return result2

def task_3():
    result3 = list(filter(lambda i: i[0][0] == 'Z', many_to_many))
    comp = result3[0][0]
    currLst = []
    for i in range(len(result3)):
        if comp == result3[i][0]:
            currLst.append(result3[i][1])
        else:
            comp = result3[i][0]
            currLst = []
            currLst.append(result3[i][1])
    return comp, currLst
```

Код программы модудьного тестирования:

```
# tdd.py
import unittest
from tasks import task_1, task_2, task_3

task_1_result = [('Inspiron', 'MultiSim', 50), ('Inspiron', 'Zoom', 100)]
task_2_result = [('Inspiron', '101.67'), ('Inspiron', '104.00'),
                 ('Inspiron', '105.00'), ('Zenbook', '110.00'),
                 ('MacBook', '115.00'), ('Inspiron', '123.33')]
task_3_result = ('Zenbook', ['Word', 'Excel', 'Chrome'])

class TasksTestCase(unittest.TestCase):
    def test_task_1(self):
        self.assertEqual(task_1_result, task_1())
    def test_task_2(self):
        self.assertEqual(task_2_result, task_2())
    def test_task_3(self):
        self.assertEqual(task_3_result, task_3())
```

Результат: