**Курзанова Анастасия ИУ5-31Б**

**Отчет по РК2, вариант 9А**

**Вариант А. Предметная область 9:**

1. «Компьютер» и «ОС» связаны соотношением один-ко-многим. Выведите список всех связанных ОС и Компьютеров, отсортированный по Компьютерам, сортировка по ОС произвольная.
2. «Компьютер» и «ОС» связаны соотношением один-ко-многим. Выведите список Компьютеров с суммарной ценой ОС в каждом Компьютере, отсортированный по суммарной цене.
3. «Компьютер» и «ОС» связаны соотношением многие-ко-многим. Выведите список всех Компьютеров, у которых в названии присутствует слово «Comp», и список ОС внутри соответствующих компьютеров.

**Задание:**

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).

**Листинг программы:**

Измененный код РК1:

from operator import itemgetter  
  
  
class OS:  
 *"""ОC"""* def \_\_init\_\_(self, id, interface, price, name, computer\_id):  
 self.id = id  
 self.interface = interface  
 self.price = price  
 self.name = name  
 self.computer\_id = computer\_id  
  
class Computer:  
 *"""Компьютер"""* def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
  
  
class OSComputer:  
 *"""ОC - Компьютер"""* def \_\_init\_\_(self, computer\_id, os\_id):  
 self.computer\_id = computer\_id  
 self.os\_id = os\_id  
  
  
"""id компьютера и его имя"""  
computers = [  
 Computer(1, 'PC-007'),  
 Computer(2, '2Comp WS-01'),  
 Computer(3, 'Admin-PS'),  
 Computer(4, '4Comp\_CU15OQA'),  
 Computer(5, 'A58CS25AD'),  
 Computer(6, '6Comp CS25'),  
]  
  
"""id ОС, тип интерфейса, цена, имя ОС,id компьютера"""  
oss = [  
 OS(1, 'Графический', 0, 'linux', 1),  
 OS(2, 'Текстовый', 0, 'Dos', 2),  
 OS(3, 'Графический', 12000, 'Windows10', 3),  
 OS(4, 'Командная строка', 0, 'Unix', 4),  
 OS(5, 'Графический', 150000, 'MacOS', 5),  
 OS(6, 'Графический', 19000, 'Windows10professional', 5),  
 OS(7, 'Графический', 8000, 'Windows7', 1),  
 OS(8, 'Текстовый', 0, 'Dos', 2),  
]  
  
"""id компьютера, id ОС"""  
oss\_computers = [  
 OSComputer(1, 1),  
 OSComputer(2, 2),  
 OSComputer(3, 3),  
 OSComputer(4, 4),  
 OSComputer(5, 5),  
 OSComputer(5, 6),  
 OSComputer(1, 7),  
 OSComputer(2, 7),  
 OSComputer(6, 3),  
]  
  
def one\_to\_many(computers, oss):  
 return [(h.name, s.name, h.interface, h.price)  
 for s in computers  
 for h in oss  
 if h.computer\_id == s.id]  
  
  
def many\_to\_many(computers, oss):  
 many\_to\_many\_temp = [(s.name, hs.computer\_id, hs.os\_id)  
 for s in computers  
 for hs in oss\_computers  
 if s.id == hs.computer\_id]  
 return [(h.id, computer\_id)  
 for name, computer\_id, os\_id in many\_to\_many\_temp  
 for h in oss if h.id == os\_id]  
  
  
def A1(computers, oss) -> list:  
 res\_31 = sorted(one\_to\_many(computers, oss), key=itemgetter(1, 0)) # sorted by street name  
 return list(res\_31)  
  
  
def A2(computers, oss) -> list:  
 res32 = []  
 for i in computers:  
 s\_oss = [ \_ for \_ in filter(lambda a: a[1]==i.name ,one\_to\_many(computers, oss) )]  
 res32.append((i.name, sum([ \_[3] for \_ in s\_oss])))  
 return sorted(res32, key=itemgetter(1, 0))  
  
  
def A3(computers, oss, str\_to\_find) -> list:  
 res33 = []  
 for i in filter(lambda a: str\_to\_find in computers[a[1]-1].name, many\_to\_many(computers, oss)):  
 res33.append((computers[i[1]-1].name, sorted([ \_.name for \_ in filter(lambda a: a.computer\_id == i[1], oss)])))  
 return sorted(res33, key=itemgetter(1, 0))  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 print('Задание А1')  
 print(A1(computers, oss))  
 print('Задание А2')  
 print(A2(computers, oss))  
 print('Задание А3')  
 print(A3(computers, oss, 'Comp'))

**Результат:**

*Задание А1*

*[('Dos', '2Comp WS-01', 'Текстовый', 0), ('Dos', '2Comp WS-01', 'Текстовый', 0), ('Unix', '4Comp\_CU15OQA', 'Командная строка', 0), ('MacOS', 'A58CS25AD', 'Графический', 150000), ('Windows10professional', 'A58CS25AD', 'Графический', 19000), ('Windows10', 'Admin-PS', 'Графический', 12000), ('Windows7', 'PC-007', 'Графический', 8000), ('linux', 'PC-007', 'Графический', 0)]*

*Задание А2*

*[('2Comp WS-01', 0), ('4Comp\_CU15OQA', 0), ('6Comp CS25', 0), ('PC-007', 8000), ('Admin-PS', 12000), ('A58CS25AD', 169000)]*

*Задание А3*

*[('6Comp CS25', []), ('2Comp WS-01', ['Dos', 'Dos']), ('2Comp WS-01', ['Dos', 'Dos']), ('4Comp\_CU15OQA', ['Unix'])]*

Process finished with exit code 0

**Код тестов:**

import unittest  
from main import Computer, OS, OSComputer, A1, A2, A3  
  
  
class RK1\_test(unittest.TestCase):  
  
 def setUp(self):  
 self.computers = [  
 # id, name  
 Computer(1, 'PC-007'),  
 Computer(2, '2Comp WS-01')  
 ]  
 self.oss = [  
 # id, type, price, name, id компьютера  
 OS(1, 'Графический', 0, 'linux', 1),  
 OS(2, 'Текстовый', 0, 'Dos', 2),  
 OS(3, 'Графический', 12000, 'Windows10', 3)  
 ]  
 # os\_id, computer\_id  
 oss\_computers = [  
 OSComputer(1, 1),  
 OSComputer(2, 2),  
 OSComputer(2, 3)  
 ]  
  
 def test\_A1(self):  
 expected\_result = [  
 ('Dos', '2Comp WS-01', 'Текстовый', 0),  
 ('linux', 'PC-007', 'Графический', 0)  
 ]  
  
 result = A1(self.computers, self.oss)  
 self.assertEqual(result, expected\_result)  
  
 def test\_A2(self):  
 expected\_result = [  
 ('2Comp WS-01', 0),  
 ('PC-007', 0)  
 ]  
 result = A2(self.computers, self.oss)  
 self.assertEqual(result, expected\_result)  
  
 def test\_A3(self):  
 expected\_result = [  
 ('2Comp WS-01', ['Dos'])  
 ]  
 result = A3(self.computers, self.oss, 'Comp')  
 self.assertEqual(result, expected\_result)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

**Результат:**

Testing started at 20:14 ...

Launching unittests with arguments python -m unittest D:/МГТУ/3 семестр/rk2/test.py in D:\МГТУ\3 семестр\rk2

Ran 3 tests in 0.003s

OK

Process finished with exit code 0