**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по рубежному контролю №2

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б  Казицин Алексей |  | преподаватель кафедры  Гапанюк Ю.Е. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2022 г.

**Условия рубежного контроля №2 по курсу БКИТ**

Рубежный контроль представляет собой разработку тестов на языке Python.

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).

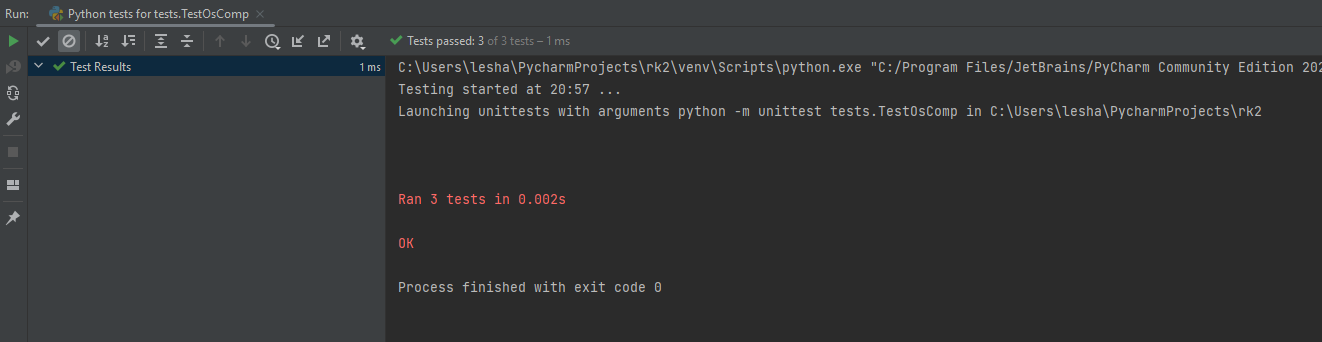
Текст программы

Main.py

from operator import itemgetter  
  
  
class OS:  
 def \_\_init\_\_(self, id, name, bit, interface, computer\_id):  
 self.id = id  
 self.name = name  
 self.bit = bit  
 self.interface = interface  
 self.computer\_id = computer\_id  
  
  
class Computer:  
 def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
  
class OSComputer:  
 def \_\_init\_\_(self, computer\_id, os\_id):  
 self.computer\_id = computer\_id  
 self.os\_id = os\_id  
  
  
computers = [  
 Computer(1, 'Vasya1'),  
 Computer(2, 'Anton-385-WH'),  
 Computer(3, 'VladVlad123'),  
 Computer(4, 'MoiComputer'),  
 Computer(5, 'Netac3405'),  
 Computer(6, 'JORDAN404-404')  
]  
  
OSs = [  
 OS(1, 'Windows10', 64, 'Graphic', 2),  
 OS(2, 'MacOS', 32, 'Graphic', 2),  
 OS(3, 'Linux', 64, 'Graphic', 4),  
 OS(4, 'DOS', 16, 'Text', 4),  
 OS(5, 'WindowsXP', 32, 'Graphic', 5),  
 OS(6, 'UNIX', 64, 'CMD', 4),  
 OS(7, 'Windows7', 64, 'Graphic', 6),  
  
]  
  
OSs\_computers = [  
 OSComputer(2, 1),  
 OSComputer(2, 2),  
 OSComputer(4, 3),  
 OSComputer(4, 4),  
 OSComputer(5, 5),  
 OSComputer(4, 6),  
 OSComputer(6, 7),  
  
 OSComputer(3, 1),  
 OSComputer(3, 2),  
 OSComputer(3, 3),  
 OSComputer(1, 4),  
 OSComputer(1, 5),  
 OSComputer(1, 6),  
 OSComputer(1, 7)  
]  
  
# Соединение данных один-ко-многим  
one\_to\_many = [(o.name, o.bit, o.interface, c.name)  
 for c in computers  
 for o in OSs  
 if o.computer\_id == c.id]  
  
# Соединение данных многие-ко-многим  
many\_to\_many\_temp = [(c.name, osc.computer\_id, osc.os\_id)  
 for c in computers  
 for osc in OSs\_computers  
 if c.id == osc.computer\_id]  
  
many\_to\_many = [(o.name, o.bit, o.interface, c\_name)  
 for c\_name, c\_id, os\_id in many\_to\_many\_temp  
 for o in OSs if o.id == os\_id]  
  
  
def task1(computers, OSs):  
 return sorted(one\_to\_many, key=itemgetter(0))  
  
  
def task2(computers, OSs):  
 second\_task = []  
  
 for c in computers:  
 c\_data = {}  
 c\_oss = list(filter(lambda i: i[3] == c.name, one\_to\_many))  
 c\_data["Name"] = c.name  
 c\_data["Number of OSs"] = len(c\_oss)  
 second\_task.append(c\_data)  
 return sorted(second\_task, key=itemgetter("Number of OSs"), reverse=True)  
  
  
def task3(computers, OSs, OSs\_computers):  
 third\_task = []  
  
 for o in OSs:  
 o\_data = {}  
 if o.bit == 32:  
 o\_data["Name"] = o.name  
 os\_comp = [mtmt[0] for mtmt in many\_to\_many\_temp if mtmt[1] == o.computer\_id]  
 o\_data["Computer"] = list(set(os\_comp))  
 o\_data["bit"] = 32  
 third\_task.append(o\_data)  
 return third\_task  
  
  
def main():  
 print('Задание №1')  
 print(  
 "Выведите список всех связанных операционных систем и компьютеров, отсортированный по операционным системам, сортировка по компьютерам произвольная.")  
 first\_task = task1(computers, OSs)  
 print(first\_task)  
  
 print('\nЗадание №2')  
 print(  
 "Выведите список компьютеров с количеством установленных операционных систем на каждом компьютере, отсортированный по количеству операционных систем.")  
 second\_task = task2(computers, OSs)  
 for i in second\_task:  
 print(i)  
  
 print('\nЗадание №3')  
 print("Выведите список всех 32-битных операционных систем, и названия компьютеров, на которых они установлены.")  
 third\_task = task3(computers, OSs, OSs\_computers)  
  
 for i in third\_task:  
 print(i, end="\n")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

Tests.py

import unittest  
from main import task1, task2, task3, OS, Computer, OSComputer  
  
  
class TestOsComp(unittest.TestCase):  
 def setUp(self):  
 self.computers = [  
 Computer(1, 'Vasya1'),  
 Computer(2, 'Anton-385-WH'),  
 Computer(3, 'VladVlad123'),  
 Computer(4, 'MoiComputer'),  
 Computer(5, 'Netac3405'),  
 Computer(6, 'JORDAN404-404')  
 ]  
  
 self.OSs = [  
 OS(1, 'Windows10', 64, 'Graphic', 2),  
 OS(2, 'MacOS', 32, 'Graphic', 2),  
 OS(3, 'Linux', 64, 'Graphic', 4),  
 OS(4, 'DOS', 16, 'Text', 4),  
 OS(5, 'WindowsXP', 32, 'Graphic', 5),  
 OS(6, 'UNIX', 64, 'CMD', 4),  
 OS(7, 'Windows7', 64, 'Graphic', 6),  
  
 ]  
  
 self.OSs\_computers = [  
 OSComputer(2, 1),  
 OSComputer(2, 2),  
 OSComputer(4, 3),  
 OSComputer(4, 4),  
 OSComputer(5, 5),  
 OSComputer(4, 6),  
 OSComputer(6, 7),  
  
 OSComputer(3, 1),  
 OSComputer(3, 2),  
 OSComputer(3, 3),  
 OSComputer(1, 4),  
 OSComputer(1, 5),  
 OSComputer(1, 6),  
 OSComputer(1, 7)  
 ]  
  
 def test\_task1(self):  
 res = task1(self.computers, self.OSs)  
 self.assertListEqual(res, [('DOS', 16, 'Text', 'MoiComputer'), ('Linux', 64, 'Graphic', 'MoiComputer'),  
 ('MacOS', 32, 'Graphic', 'Anton-385-WH'), ('UNIX', 64, 'CMD', 'MoiComputer'),  
 ('Windows10', 64, 'Graphic', 'Anton-385-WH'),  
 ('Windows7', 64, 'Graphic', 'JORDAN404-404'),  
 ('WindowsXP', 32, 'Graphic', 'Netac3405')])  
  
 def test\_task2(self):  
 res = task2(self.computers, self.OSs)  
 self.assertListEqual(res, [{'Name': 'MoiComputer', 'Number of OSs': 3},  
 {'Name': 'Anton-385-WH', 'Number of OSs': 2},  
 {'Name': 'Netac3405', 'Number of OSs': 1},  
 {'Name': 'JORDAN404-404', 'Number of OSs': 1},  
 {'Name': 'Vasya1', 'Number of OSs': 0},  
 {'Name': 'VladVlad123', 'Number of OSs': 0}])  
  
 def test\_task3(self):  
 res = task3(self.computers, self.OSs, self.OSs\_computers)  
 self.assertListEqual(res, [  
 {'Name': 'MacOS', 'Computer': ['Anton-385-WH'], 'bit': 32},  
 {'Name': 'WindowsXP', 'Computer': ['Netac3405'], 'bit': 32}])  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 unittest.main()

Результат работы программы