**Московский государственный технический**

**Университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Базовые компоненты интернет-технологий» Отчет по Рубежному Контролю №2

«Изучение основных конструкций языка Python»

Выполнил:

студент группы ИУ5-33Б

Дувакин А.В.

Проверил:

Нардид А.Н.

2023 г.

# 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).

# Текст программы

**Файл main.py:**

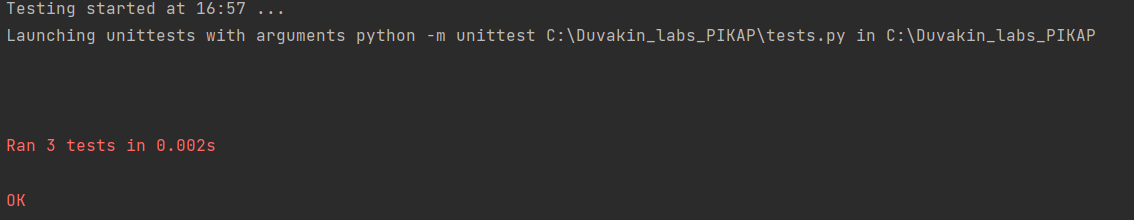
from operator import itemgetter  
class Browser:  
 def \_\_init\_\_(self, id, name, size, comp\_id):  
 self.id = id  
 self.name = name  
 self.size = size  
 self.comp\_id = comp\_id  
  
class Computer:  
 def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
class BrowComp:  
  
 def \_\_init\_\_(self, comp\_id, brow\_id):  
 self.comp\_id = comp\_id  
 self.brow\_id = brow\_id  
  
computers = [  
 Computer(1, 'Enigma'),  
 Computer(2, 'Macintosh'),  
 Computer(3, 'Agat'),  
 Computer(4, 'Macintosh'),  
 Computer(11, 'Asus'),  
 Computer(22, 'Lenovo'),  
 Computer(33, 'Apple'),  
 Computer(44, 'AMD'),  
]  
browsers = [  
 Browser(1, 'Chrome', 12300, 1),  
 Browser(2, 'Opera', 34500, 2),  
 Browser(3, 'Safari', 56400, 3),  
 Browser(4, 'Yandex', 64400, 3),  
 Browser(5, 'OperaGX', 75600, 4),  
]  
  
browsers\_computers = [  
 BrowComp(1, 1), BrowComp(2, 2), BrowComp(3, 3), BrowComp(3, 4), BrowComp(4, 5),  
 BrowComp(11, 1), BrowComp(22, 2), BrowComp(33, 3), BrowComp(33, 4), BrowComp(44, 5),  
]  
  
def n1\_sol(o\_to\_m):  
  
 numb\_1 = sorted([(name, size, name) for name, size, name in o\_to\_m if name.startswith('A')], key=itemgetter(2))  
 for i in numb\_1:  
 return(i)  
def n2\_sol(o\_to\_m):  
 numb\_2\_unsorted = []  
 for d in computers:  
 d\_emps = list(filter(lambda i: i[2] == d.name, o\_to\_m))  
 if len(d\_emps) > 0:  
 d\_sizes = [size for \_, size, \_ in d\_emps]  
 d\_sizes\_min = min(d\_sizes)  
 numb\_2\_unsorted.append((d.name, d\_sizes\_min))  
 numb\_2 = sorted(numb\_2\_unsorted, key=itemgetter(1), )  
 return(numb\_2)  
  
def n3\_sol(m\_to\_m):  
 numb\_3 = sorted(m\_to\_m, key=itemgetter(0))  
 return(numb\_3)  
  
def main():  
 o\_to\_m = [(brw.name, brw.size, cmp.name)  
 for cmp in computers  
 for brw in browsers  
 if brw.comp\_id == cmp.id]  
  
 m\_to\_m\_tmp = [(cmp.name, brwcmp.comp\_id, brwcmp.brow\_id)  
 for cmp in computers  
 for brwcmp in browsers\_computers  
 if cmp.id == brwcmp.comp\_id]  
  
 m\_to\_m = [(brw.name, brw.size, tmp\_name)  
 for tmp\_name, tmp\_id, emp\_id in m\_to\_m\_tmp  
 for brw in browsers if brw.id == emp\_id]  
  
  
 print('N 1')  
 print(n1\_sol(o\_to\_m))  
  
  
 print('N 2')  
 print(n2\_sol(o\_to\_m))  
  
 print('N 3')  
 print (n3\_sol(m\_to\_m))  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

**Файл tests.py:**

import unittest  
from main import \*  
  
  
class TestRK2(unittest.TestCase):  
 # Компьютеры  
 computers = [  
 Computer(1, 'Enigma'),  
 Computer(2, 'Macintosh'),  
 Computer(3, 'Agat'),  
 Computer(4, 'Macintosh'),  
 Computer(11, 'Asus'),  
 Computer(22, 'Lenovo'),  
 Computer(33, 'Apple'),  
 Computer(44, 'AMD'),  
 ]  
  
 # Микропроцессоры  
 browsers = [  
 Browser(1, 'Chrome', 12300, 1),  
 Browser(2, 'Opera', 34500, 2),  
 Browser(3, 'Safari', 56400, 3),  
 Browser(4, 'Yandex', 64400, 3),  
 Browser(5, 'OperaGX', 75600, 4),  
 ]  
 browsers\_computers = [  
 BrowComp(1, 1), BrowComp(2, 2), BrowComp(3, 3), BrowComp(3, 4), BrowComp(4, 5),  
 BrowComp(11, 1), BrowComp(22, 2), BrowComp(33, 3), BrowComp(33, 4), BrowComp(44, 5),  
 ]  
  
 def test\_A1(self):  
 o\_to\_m = [(brw.name, brw.size, cmp.name)  
 for cmp in computers  
 for brw in browsers  
 if brw.comp\_id == cmp.id]  
 self.assertEqual(n1\_sol(o\_to\_m),  
 ('Agat', 56400, 'Agat'))  
  
 def test\_A2(self):  
 o\_to\_m = [(brw.name, brw.size, cmp.name)  
 for cmp in computers  
 for brw in browsers  
 if brw.comp\_id == cmp.id]  
 self.assertEqual(n2\_sol(o\_to\_m),  
 [('Enigma', 12300)])  
  
 def test\_A3(self):  
 m\_to\_m\_tmp = [(cmp.name, brwcmp.comp\_id, brwcmp.brow\_id)  
 for cmp in computers  
 for brwcmp in browsers\_computers  
 if cmp.id == brwcmp.comp\_id]  
  
 m\_to\_m = [(brw.name, brw.size, tmp\_name)  
 for tmp\_name, tmp\_id, emp\_id in m\_to\_m\_tmp  
 for brw in browsers if brw.id == emp\_id]  
 self.assertEqual(n3\_sol(m\_to\_m),  
 [('Chrome', 12300, 'Enigma'),  
 ('Chrome', 12300, 'Asus'), ('Opera', 34500, 'Macintosh'), ('Opera', 34500, 'Lenovo'),  
 ('OperaGX', 75600, 'Macintosh'), ('OperaGX', 75600, 'AMD'), ('Safari', 56400, 'Agat'),  
 ('Safari', 56400, 'Apple'), ('Yandex', 64400, 'Agat'), ('Yandex', 64400, 'Apple')])  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

**Результаты тестов:**

Положительный результат:



Допущена ошибка:

