

**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:

```
Testing started at 16:57 ...
Launching unittests with arguments python -m unittest C:\Duvakin_labs_PIKAP\tests.py in C:\Duvakin_labs_PIKAP

Ran 3 tests in 0.002s

OK
```

Допущена ошибка:

```
Testing started at 17:58 ...
Launching unittests with arguments python -m unittest C:\Duvakin_labs_PIKAP\tests.py in C:\Duvakin_labs_PIKAP

[('Enigma', 1230)] != [('Enigma', 12300)]

Expected : [('Enigma', 12300)]
Actual   : [('Enigma', 1230)]
<Click to see difference>

Traceback (most recent call last):
  File "C:\Duvakin_labs_PIKAP\tests.py", line 44, in test_A2
    self.assertEqual(n2_sol(o_to_m),
AssertionError: Lists differ: [('Enigma', 12300)] != [('Enigma', 1230)]

First differing element 0:
('Enigma', 12300)
('Enigma', 1230)

- [('Enigma', 12300)]
?                  -

+ [('Enigma', 1230)]
```