



**Министерство науки и высшего образования Российской
Федерации Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

**Рубежный контроль №2
по дисциплине «Базовые компоненты интернет-технологий»**

**Выполнил:
студент группы ИУ5-33Б
Некрасов С. А.**

**Проверил:
Канев А.И.**

2021 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 Os:
    def __init__(self, id, name, cost, comp_id):
        self.id = id
        self.name = name
        self.cost = cost
        self.comp_id = comp_id

class Computer:
    def __init__(self, id, name):
        self.id = id
        self.name = name

class OsComp:
    def __init__(self, os_id, comp_id):
        self.os_id = os_id
        self.comp_id = comp_id

comps = [
    Computer(1, 'Acer'),
    Computer(2, 'Msi'),
    Computer(3, 'Asus'),
    Computer(4, 'Lenovo'),
    Computer(5, 'Apple')
]

oses = [
    Os(1, 'Windows', 2900, 3),
    Os(2, 'Abuntu', 3510, 2),
    Os(3, 'MacOS', 5450, 5),
    Os(4, 'ChromeOS', 1480, 2),
    Os(5, 'Adora', 4370, 3)
]

oses_comps = [
    OsComp(1, 3),
    OsComp(2, 2),
    OsComp(3, 5),
    OsComp(4, 2),
    OsComp(5, 3),

    OsComp(1, 2),
    OsComp(2, 4),
    OsComp(3, 1),
    OsComp(4, 4),
```

```

        OsComp(5, 1)
    ]

def test1(one_to_many):
    res1 = []
    for name, _, comp in one_to_many:
        if name[0] == "A":
            res1.append((name, comp))
    return res1

def test2(one_to_many):
    res2 = []
    for c in comps:
        c_oses = list(filter(lambda x: x[2] == c.name, one_to_many))
        if len(c_oses) > 0:
            c_cost = [cost for _, cost, _ in c_oses]
            c_cost_min = min(c_cost)
            res2.append((c.name, c_cost_min))
    res2_sorted = sorted(res2, key=itemgetter(1))
    return res2_sorted

def test3(many_to_many):
    res3 = []
    for name, _, comp in many_to_many:
        res3.append((name, comp))
    res3_sorted = sorted(res3, key=itemgetter(0))
    return res3_sorted

def main():
    one_to_many = [(o.name, o.cost, c.name)
                    for c in comps
                    for o in oses
                    if c.id == o.comp_id]

    many_to_many_temp = [(c.name, oc.os_id, oc.comp_id)
                          for c in comps
                          for oc in oses_comps
                          if c.id == oc.comp_id]

    many_to_many = [(o.name, o.cost, c_name)
                     for c_name, oc_id, c_id in many_to_many_temp
                     for o in oses
                     if o.id == oc_id]

    print('\nЗадание B1:\n', test1(one_to_many))
    print('\nЗадание B2:\n', test2(one_to_many))
    print('\nЗадание B3:\n', test3(many_to_many))

if __name__ == "__main__":
    main()

```

Файл test.py

```

import unittest # автоматизация тестов
import sys, os # предоставляет системе особые параметры и функции

from main import *

```

```

sys.path.append(os.getcwd()) # добавить путь поиска модулей

class Test(unittest.TestCase):
    def test1(self):
        one_to_many = [(o.name, o.cost, c.name)
                        for c in comps
                        for o in oses
                        if c.id == o.comp_id]
        self.assertEqual(test1(one_to_many), [('Abuntu', 'Msi'), ('Adora',
'Asus')])

    def test2(self):
        one_to_many = [(o.name, o.cost, c.name)
                        for c in comps
                        for o in oses
                        if c.id == o.comp_id]
        self.assertEqual(test2(one_to_many), [('Msi', 1480), ('Asus', 2900),
('Apple', 5450)])

    def test3(self):
        many_to_many_temp = [(c.name, oc.os_id, oc.comp_id)
                              for c in comps
                              for oc in oses_comps
                              if c.id == oc.comp_id]
        many_to_many = [(o.name, o.cost, c_name)
                         for c_name, oc_id, c_id in many_to_many_temp
                         for o in oses
                         if o.id == oc_id]
        self.assertEqual(test3(many_to_many),
        [('Abuntu', 'Msi'), ('Abuntu', 'Lenovo'), ('Adora',
'Acer'), ('Adora', 'Asus'),
        ('ChromeOS', 'Msi'), ('ChromeOS', 'Lenovo'),
        ('MacOS', 'Acer'), ('MacOS', 'Apple'),
        ('Windows', 'Msi'), ('Windows', 'Asus')])

```

Результат выполнения:

```

C:\Users\79508\PycharmProjects\RK2\venv\Scripts\python.
Testing started at 19:28 ...

```

```

Ran 3 tests in 0.002s

```

```

OK

```