

**Московский государственный технический  
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «БКИТ»  
Отчет по рубежному контролю №2

Выполнил:

студент группы ИУ5-35Б  
Крылов Дмитрий

Подпись и дата:

Проверил:

преподаватель каф. ИУ5  
Гапанюк Ю. Е.

Подпись и дата:

Москва, 2022 г.

## Описание задания

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).

Вариант 9

9	Операционная система	Компьютер
---	----------------------	-----------

# Текст программы

## main.py

```
from operator import itemgetter

class OpSys:
    """Операционная система"""
    def __init__(self, id, fio, size, pc_id):
        self.id = id
        self.fio = fio
        self.size = size
        self.pc_id = pc_id

class Pc:
    """Компьютер"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class OpSys_s:
    """
    'Операционные системы' для реализации
    связи многие-ко-многим
    """
    def __init__(self, pc_id, os_id):
        self.os_id = os_id
        self.pc_id = pc_id

# Компьютеры
Pcs = [
    Pc(1, 'Игровой компьютер'),
    Pc(2, 'Рабочий компьютер'),
    Pc(3, 'Офисный ноутбук'),
]

# Операционные системы
# Номер / название / количество занимаемой памяти (чистый вес ОС в мб) / установлен на ПК
OSs = [
    OpSys(1, 'MacOs-ов', 32000, 1),
    OpSys(2, 'Windows', 24000, 1),
    OpSys(3, 'Harmony-ов', 41000, 2),
    OpSys(4, '(Linux) Raspbian', 12000, 2),
    OpSys(5, '(Linux) Ubuntu', 15000, 3),
    OpSys(6, '(Linux) Kali-ов', 17000, 3),
]

os_pk = [
    OpSys_s(1, 1),
    OpSys_s(2, 1),
    OpSys_s(3, 2),
    OpSys_s(4, 2),
    OpSys_s(5, 3),
    OpSys_s(6, 3),
]
```

```

def task1():
    global one_to_many, many_to_many
    result = []

    for i in range(len(one_to_many)):
        if one_to_many[i][0][-2:]=="ов":
            result.append((one_to_many[i][0], one_to_many[i][2]))
    return result

def task2():
    global one_to_many, many_to_many
    result = []

    arr = arr1 = []
    for x in PCs:
        # Список ОС
        d_muss = list(filter(lambda i: i[2] == x.name, one_to_many))
        if len(d_muss) > 0:
            # занимаемая память
            d_sals = [sal for _, sal, _ in d_muss]
            # Средний объем занимаемой ОС-ми памяти
            d_sals_sum = sum(d_sals)
            arr.append((x.name, d_sals_sum/len(d_muss)))
            arr1=sorted(arr, key=itemgetter(1), reverse=True)
    for i in arr1:
        result.append((i[0],i[1].__round__()))
    return result

def task3():
    global one_to_many, many_to_many
    result = {}

    arr2 = []
    for d in PCs:
        if 'И' in d.name:
            arr2.append(d.id)

    for x in range(int(len(arr2))):
        result[many_to_many[x][2]] = []
        for i in OSs:
            if i.pc_id == arr2[x]:
                result[many_to_many[x][2]].append(i.fio)
                #print(i.fio)

    return result

global one_to_many, many_to_many
"""Основная функция"""
one_to_many = [(e.fio, e.size, d.name) for d in PCs for e in OSs if e.pc_id == d.id]
many_to_many_temp = [(d.name, ed.pc_id, ed.pc_id) for d in PCs for ed in os_pk if d.id == ed.pc_id]
many_to_many = [(e.fio, e.size, orc_name) for orc_name, orc_id, mus_id in many_to_many_temp for e in OSs if e.id == mus_id ]

def main():
    print('Задание A1')
    print(task1())
    print('\nЗадание A2')
    print(task2())
    print('\nЗадание A3')
    print(task3())

if __name__ == '__main__':
    main()

```

## tdd2.py

```
import unittest
from main2 import *

task1_result = [('MacOs-ов', 'Игровой компьютер'), ('Harmony-ов', 'Рабочий компьютер'), ('(Linux) Kali-ов', 'Офисный ноутбук')]
task2_result = [('Игровой компьютер', 28000), ('Рабочий компьютер', 26500), ('Офисный ноутбук', 16000)]
task3_result = {'Игровой компьютер': ['MacOs-ов', 'Windows']}

class TestEquation(unittest.TestCase):
    def test_check_task1(self):
        self.assertEqual(task1_result, task1())

    def test_check_task2(self):
        self.assertEqual(task2_result, task2())

    def test_check_task3(self):
        self.assertEqual(task3_result, task3())

if __name__ == '__main__':
    unittest.main()
```

## Результаты выполнения программы

### main.py

```
Задание A1
[('MacOs-ов', 'Игровой компьютер'), ('Harmony-ов', 'Рабочий компьютер'), ('(Linux) Kali-ов', 'Офисный ноутбук')]
```

Задание A2

```
[('Игровой компьютер', 28000), ('Рабочий компьютер', 26500), ('Офисный ноутбук', 16000)]
```

Задание A3

```
{'Игровой компьютер': ['MacOs-ов', 'Windows']}
```

### tdd2.py

```
Testing started at 12:13 ...
Launching unittests with arguments python -m unittest /Users/dmitrykrylov/PycharmProjects/rk1/tdd2.py in /Users/dmitrykrylov/PycharmProjects/rk1
```

Ran 3 tests in 0.001s

OK

Process finished with exit code 0