



**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»**

Вариант Д-9

**Выполнил:
студент группы ИУ5-35Б
Костин С.С.**

2021 г.

Задание РК-1:

Вариант Д.

«Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.

«Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате (отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений).

«Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.

ПО Вариант №9

Операционная система

Компьютер

Задание РК-2:

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

Файл RK.py:

```
from operator import itemgetter

class OS:
    """Операционная система"""
    def __init__(self, id, name, developer):
        self.id = id
        self.name = name
        self.developer = developer

class Computer:
    """Компьютер"""
    def __init__(self, id, owner, price, OS_id):
        self.id = id
        self.owner = owner
        self.price = price
        self.OS_id = OS_id

class CompOS:
    """Связь ОС-компьютер"""
    def __init__(self, comp_id, OS_id):
        self.OS_id = OS_id
```

```

        self.comp_id = comp_id

# Операционные системы
OSs = [
    OS(1, 'Windows', 'Microsoft'),
    OS(2, 'Linux', 'Линус'),
    OS(3, 'macOS', 'Apple'),
    OS(4, 'Альфа-ОС', 'Петров'),
]

# Сотрудники
comps = [
    Computer(1, 'Кощеев', 66000, 1),
    Computer(2, 'Сидоров', 110000, 1),
    Computer(3, 'Николаев', 56000, 2),
    Computer(4, 'Афанасьев', 120000, 3),
    Computer(5, 'Петров', 33000, 4),
    Computer(6, 'Остров', 58000, 2),
]

# Дополнительные ОС Компьютеров
cmpOSes = [
    CompOS(1, 1),
    CompOS(2, 1),
    CompOS(3, 2),
    CompOS(4, 3),
    CompOS(5, 4),
    CompOS(6, 2),

    CompOS(4, 1),
    CompOS(4, 4),
    CompOS(1, 4),
    CompOS(5, 2),
]

def CompByID(id):
    return list(filter(lambda c: c.id == id, comps))[0]

def OSByID(id):
    return list(filter(lambda c: c.id == id, OSs))[0]

def task1():
    print("Задание Д1")
    relations = getRelations_OtO()
    res_d1 = list(filter(lambda x: x[0][len(x[0])-2:] == 'ов', relations))
    print(res_d1)
    return res_d1

def task2():
    print("Задание Д2")
    relations = getRelations_OtO()
    res_d2_unsorted = list()
    # Для всех ОС
    for os in OSs:
        # Получить компьютеры на этой ОС
        compsWithOS = list(filter(lambda x: x[2] == os.name, relations))
        # Если таковые имеются
        if len(compsWithOS) > 0:
            # Собираем все стоимости компов
            _prices = [price for _, price, _ in compsWithOS]
            # Суммируем и ищем среднее
            aver = sum(_prices) / len(_prices)
            res_d2_unsorted.append((os.name, aver))
    # Сортировка по среднему
    res_d2 = sorted(res_d2_unsorted, key=itemgetter(1), reverse=True)
    print(res_d2)

```

```

return res_d2

def task3():
    print("Задание Д3")
    res_d3 = dict()
    # для каждой связи
    for link in cmpOSes:
        # имя ОС
        currentName = OSByID(link.OS_id).name
        if (currentName[0] == 'А'):
            if currentName not in res_d3:
                res_d3[currentName] = list()
            # добавить фамилию владельца
            res_d3[currentName].append(CompByID(link.comp_id).owner)
    print(res_d3)
    return res_d3

def getRelations_OtO():
    one_to_many = [(c.owner, c.price, os.name)
                    for c in comps
                    for os in OSs
                    if c.OS_id==os.id]
    return one_to_many

if __name__ == '__main__':
    # Соединение данных один-ко-многим
    task1()
    task2()
    task3()

```

Файл tdd.py:

```

import unittest
import RK

class TestCompOS(unittest.TestCase):
    def test_t1(self):
        excpRes = [('Сидоров', 110000, 'Windows'), ('Петров', 33000, 'Альфа-ОС'), ('Остров', 58000, 'Linux')]
        self.assertEqual(RK.task1(), excpRes)

    def test_t2(self):
        excpRes = [('macOS', 120000.0), ('Windows', 88000.0), ('Linux', 57000.0), ('Альфа-ОС', 33000.0)]
        self.assertEqual(RK.task2(), excpRes)

    def test_t3(self):
        excpRes = {'Альфа-ОС': ['Петров', 'Афанасьев', 'Кощеев']}
        self.assertEqual(RK.task3(), excpRes)

if __name__ == "__main__":
    unittest.main()

```

Результаты работы

Файл RK.py:

```
Задание Д1
[('Сидоров', 110000, 'Windows'), ('Петров', 33000, 'Альфа-ОС'),
 ('Остров', 58000, 'Linux')]
Задание Д2
[('macOS', 120000.0), ('Windows', 88000.0), ('Linux', 57000.0),
 ('Альфа-ОС', 33000.0)]
Задание Д3
{'Альфа-ОС': ['Петров', 'Афанасьев', 'Кощеев']}
Press any key to continue . . .
```

Файл tdd.py:

```
Задание Д1.
[('Сидоров', 110000, 'Windows'), ('Петров', 33000, 'Альфа-ОС'), ('Остров', 58000, 'Linux')]
Задание Д2
[('macOS', 120000.0), ('Windows', 88000.0), ('Linux', 57000.0), ('Альфа-ОС', 33000.0)]
Задание Д3
{'Альфа-ОС': ['Петров', 'Афанасьев', 'Кощеев']}.
-----
Ran 3 tests in 0.003s
OK
Press any key to continue . . . █
```