



**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»**

Лабораторная работа №1  
по дисциплине «Электротехника» на тему:  
«Цепи постоянного тока»  
Вариант №17

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

Проверил:  
Белодедов М.В.

2021 г.

## Текст программы

```
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
compOSes = [
    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 main():
```

```

"""Основная функция"""
# Соединение данных один-ко-многим
one_to_many = [(c.owner, c.price, os.name)
                for c in comps
                for os in OSs
                if c.OS_id==os.id]

print('Задание Д1')
res_d1 = list(filter(lambda x: x[0][len(x[0])-2:]=='ов', one_to_many))
print(res_d1)

print('Задание Д2')
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, one_to_many))
    # 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)

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)

if __name__ == '__main__':
    main()

```

## Результаты работы

```

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 . . .

```