

Государственное образовательное учреждение высшего профессионального образования



«Московский государственный технический университет  
имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Системы обработки информации и управления» (ИУ-5)

## РУБЕЖНЫЙ КОНТРОЛЬ

№ 2

Группа ИУ5-35Б

Студент \_\_\_\_\_ 16.12.2024 /А. А. Торопыгин/

(Подпись, дата)

(И.О.Фамилия)

Преподаватель \_\_\_\_\_ /Ю. Е. Гапанюк/

(Подпись, дата)

(И.О.Фамилия)

2024

## Текст программы:

```
from operator import itemgetter

def unit_test(**kwargs):
    def inner_decorator(func):
        def wrapped(*args):
            assert 'output' in kwargs

            result = func()
            assert result == kwargs['output'], \
                f'incorrect return from function {func.__name__}()\n\
                recieved: {result}\n\
                expected: {kwargs['output']}'
            print(f'\nfunction \'{func.__name__}\'' test completed
                successfully\n')

            return func(*args)
        return wrapped
    return inner_decorator

class Driver:
    """Водитель"""
    def __init__(self, id, fio, exp, ap_id):
        self.id = id
        self.fio = fio
        self.exp = exp #стаж в годах
        self.ap_id = ap_id

class AP:
    """Автопарк"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class DriverAP:
    """
    'Водители автопарка' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """
```

```

    def __init__(self, ap_id, driver_id):
        self.ap_id = ap_id
        self.driver_id = driver_id

# Автопарки
aps = [
    AP(1, 'Транспортный Альянс'),
    AP(2, 'ЭкоДрайв'),
    AP(3, 'АвтоМир')
]

# Водители
drivers = [
    Driver(1, 'Алексеев', 10, 1),
    Driver(2, 'Борисов', 11, 1),
    Driver(3, 'Васильев', 15, 2),
    Driver(4, 'Герасимов', 30, 2),
    Driver(5, 'Данилов', 13, 3)
]

drivers_aps = [
    DriverAP(1, 1),
    DriverAP(1, 2),
    DriverAP(1, 3),

    DriverAP(2, 3),
    DriverAP(2, 4),
    DriverAP(2, 5),

    DriverAP(3, 1),
    DriverAP(3, 2),
    DriverAP(3, 4),
    DriverAP(3, 5)
]

# Соединение данных один-ко-многим
one_to_many = [(d.fio, d.exp, ap.name)
                for ap in aps
                for d in drivers]

```

```

        if d.ap_id==ap.id]

@unit_test(
    output = [
        ('Борисов', 'Транспортный Альянс'),
        ('Герасимов', 'ЭкоДрайв'),
        ('Данилов', 'АвтоМир')
    ]
)

def drivers_end_with_ov():
    res_1 = [(i[0], i[2]) for i in one_to_many if i[0].endswith("ов")]
    return res_1

@unit_test(
    output = [
        ('ЭкоДрайв', 22.5),
        ('АвтоМир', 13.0),
        ('Транспортный Альянс', 10.5)
    ]
)

def sort_by_mid_exp():
    res_2_unsorted = []
    for ap in aps:
        ap_drivers = list(filter(lambda i: i[2]==ap.name, one_to_many)) #
получаем список водителей
        if len(ap_drivers) > 0:
            ap_exps = [exp for _,exp,_ in ap_drivers] # получаем список
зарплат
            ap_exps_mid = sum(ap_exps)/len(ap_exps) # вычисляем среднюю
з/п
            res_2_unsorted.append((ap.name, ap_exps_mid))
    res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True) #
сортируем по убыванию средней з/п
    return res_2

# Соединение данных многие-ко-многим
many_to_many_temp = [(ap.name, dap.ap_id, dap.driver_id)
    for ap in aps
    for dap in drivers_aps
    if ap.id==dap.ap_id]

```

```

many_to_many = [(d.fio, d.exp, dep_name)
    for dep_name, ap_id, driver_id in many_to_many_temp
    for d in drivers if d.id==driver_id]

@unit_test(
    output = {
        'АвтоМир': ['Алексеев', 'Борисов', 'Герасимов', 'Данилов']
    }
)

def ap_starts_with_a():
    res_3 = {}
    for ap in aps:
        if ap.name.startswith("A"):
            # получаем все записи о водителях автопарка
            ap_drivers = list(filter(lambda i: i[2]==ap.name, many_to_many))
            # получаем список имен водителей
            ap_drivers_names = [x for x,_,_ in ap_drivers]
            res_3[ap.name] = ap_drivers_names
    return res_3

if __name__ == '__main__':
    print('Задание Д1\n', drivers_end_with_ov())
    print('Задание Д2\n', sort_by_mid_exp())
    print('Задание Д3\n', ap_starts_with_a())

```

## Вывод:

```

• andrey@andrey-HP-EliteBook-840-G5:~/BKIT$ /bin/python3 /home/andrey/BKIT/RK_1/main.py

function 'drivers_end_with_ov' test completed successfully

Задание Д1
[('Борисов', 'Транспортный Альянс'), ('Герасимов', 'ЭкоДрайв'), ('Данилов', 'АвтоМир')]

function 'sort_by_mid_exp' test completed successfully

Задание Д2
[('ЭкоДрайв', 22.5), ('АвтоМир', 13.0), ('Транспортный Альянс', 10.5)]

function 'ap_starts_with_a' test completed successfully

Задание Д3
{'АвтоМир': ['Алексеев', 'Борисов', 'Герасимов', 'Данилов']}

```