



**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 №1
по дисциплине «Базовые компоненты интернет-технологий»
Вариант №В3**

**Выполнила:
студентка группы ИУ5-33Б
Валова С. В.**

**Проверил:
Преподаватель кафедры ИУ-5
Гапанюк Ю. Е.**

2022 г.

Условие варианта В для предметной области «Водитель-Автопарк»

1. «Водитель» и «Автопарк» связаны соотношением один-ко-многим. Выведите список всех водителей, у которых фамилия начинается с буквы «А», и названия их автопарков.
2. «Водитель» и «Автопарк» связаны соотношением один-ко-многим. Выведите список автопарков с минимальной зарплатой водителей в каждом автопарке, отсортированный по минимальной зарплате.
3. «Водитель» и «Автопарк» связаны соотношением многие-ко-многим. Выведите список всех связанных водителей и автопарков, отсортированный по водителям, сортировка по автопаркам произвольная.

Текст программы

```
#Водитель Автопарк
from operator import itemgetter

class Driver:
    def __init__(self, id, fio, pay, id autopark):
        self.id = id
        self.fio = fio
        self.pay = pay
        self.IdAutopark = id_autopark

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

class Ap_to_Dr:
    def __init__(self, id_Ap, id_Dr):
        self.id_Ap = id_Ap
        self.id_Dr = id_Dr

Drivers = [
    Driver(1, 'Артаев', 20000, 1),
    Driver(2, 'Алешин', 22000, 2),
    Driver(3, 'Светличко', 20000, 1),
    Driver(4, 'Кунцевский', 25000, 3),
    Driver(5, 'Аверьянов', 26000, 3),
    Driver(6, 'Приходько', 21000, 1)
]

Autoparks = [
    Autopark(1, 'М-Такси'),
    Autopark(2, 'БасЭлитСервис'),
    Autopark(7, 'Авангард'),
    Autopark(3, 'Яндекс-такси'),
    Autopark(12, 'ТК Повозкин')
]

Aps_Drs = [
    Ap_to_Dr(1, 1),
    Ap_to_Dr(2, 2),
    Ap_to_Dr(1, 3),
    Ap_to_Dr(3, 4),
    Ap_to_Dr(3, 5),
```

```

        Ap_to_Dr(4, 6),
        Ap_to_Dr(1, 7)
    ]

def main1():
    one_to_many = [(i.fio, i.pay, j.name)
                    for i in Drivers
                    for j in Autoparks
                    if i.IdAutopark == j.id]

    many_to_many_temp = [(a.name, n.id_Ap, n.id_Dr)
                          for a in Autoparks
                          for n in Aps_Drs
                          if a.id == n.id_Ap]

    many_to_many = [(i.fio, i.pay, Ap_name)
                    for Ap_name, id_Ap, id_Dr in many_to_many_temp
                    for i in Drivers
                    if i.id == id_Dr]

    print('Задание B1')
    result1 = list(filter(lambda i: i[0][0] == "A", one_to_many))
    print(result1)

    print('Задание B2')
    list_of_parks = []
    result2 = []
    for i in one_to_many:
        if i[2] not in list_of_parks:
            list_of_parks.append(i[2])
    for j in list_of_parks:
        min = 1000000000
        for i in one_to_many:
            if j == i[2] and i[1] < min:
                min = i[1]
        result2.append((j, min))
    print(result2)

    print('Задание B3')
    result3 = sorted(one_to_many, key=itemgetter(0))
    for i in result3:
        print(i)

if __name__ == '__main__':
    main1()

```

Результат выполнения:

```

Задание B1
[('Артаев', 20000, 'М-Такси'), ('Алешин', 22000, 'БасЭлитСервис'), ('Аверьянов', 26000, 'Яндекс-такси')]
Задание B2
[('М-Такси', 20000), ('БасЭлитСервис', 22000), ('Яндекс-такси', 25000)]
Задание B3
('Аверьянов', 26000, 'Яндекс-такси')
('Алешин', 22000, 'БасЭлитСервис')
('Артаев', 20000, 'М-Такси')
('Кунцевский', 25000, 'Яндекс-такси')
('Приходько', 21000, 'М-Такси')
('Светличко', 20000, 'М-Такси')

Process finished with exit code 0

```