

МГТУ им. Н.Э. Баумана
Кафедра «Системы обработки информации и
управления»

Рубежный контроль №1
«Базовые компоненты интернет-технологий»

Студент группы ИУ5-34Б:

Гордеев Матвей Владиславович

Преподаватель кафедры ИУ5:

Гапанюк Юрий Евгеньевич

Москва, 2022

Вариант Г. Предметная область 3.

1. «Водитель» и «Автопарк» связаны соотношением один-ко-многим. Выведите список всех автопарков, у которых название начинается с буквы «В», и список работающих в них водителей.
2. «Водитель» и «Автопарк» связаны соотношением один-ко-многим. Выведите список автопарков с максимальным рейтингом водителей в каждом автопарке, отсортированный по максимальному рейтингу.
3. «Водитель» и «Автопарк» связаны соотношением многие-ко-многим. Выведите список всех связанных водителей и автопарков, отсортированный по автопаркам, сортировка водителей по длине имени.

Листинг программы:

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

''' водитель '''
class Driver:
    def __init__(self, id, name, Rate, Autopark_id):
        self.id = id
        self.name = name
        self.Autopark_id = Autopark_id
        self.Rate=Rate

''' связь водителей с автопарками '''
class DriverInAutopark:
    def __init__(self, Driver_id, Autopark_id):
        self.Driver_id = Driver_id
        self.Autopark_id = Autopark_id

Autoparks = [
    Autopark(1, "Citymobile"),
    Autopark(2, "BestTaxi"),
    Autopark(3, "Blablacar"),
    Autopark(11, "AutoFree"),
    Autopark(22, "Uber"),
    Autopark(33, "AutoForYou"),
]

Drivers = [
    Driver(1, "Mirshin", 3.0, 1),
    Driver(2, "Samsonov", 5.0, 2),
    Driver(3, "Victorovich", 2.1, 3),
    Driver(4, "Alexandrovich", 4.1, 3),
    Driver(5, "Grigoryevich", 1.4, 3)
]
```

```

DriversInAutoparks = [
    DriverInAutopark(1, 1),
    DriverInAutopark(2, 2),
    DriverInAutopark(3, 3),
    DriverInAutopark(3, 4),
    DriverInAutopark(3, 5),
    DriverInAutopark(11, 1),
    DriverInAutopark(22, 2),
    DriverInAutopark(33, 3),
    DriverInAutopark(33, 4),
    DriverInAutopark(33, 5)
]

def main():
    ''' СВЯЗЬ ОДИН-КО-МНОГУМ'''
    one_to_many = [(dr.name, dr.Rate, au.name)
                    for au in Autoparks
                    for dr in Drivers
                    if dr.Autopark_id == au.id]

    many_to_many_temp = [(au.name, DrInAu.Autopark_id, DrInAu.Driver_id)
                          for au in Autoparks
                          for DrInAu in DriversInAutoparks
                          if au.id == DrInAu.Autopark_id]

    ''' СВЯЗЬ МНОГО-КО-МНОГИМ'''
    many_to_many = [(dr.name, dr.Rate, AutoparkName)
                     for AutoparkName, AutoparkId, DriverId in many_to_many_temp
                     for dr in Drivers
                     if dr.id == DriverId]

    print('one_to_many')
    print(one_to_many)
    print('many_to_many_temp')
    print(many_to_many_temp)

```

```

print('many_to_many')
print(many_to_many)

'''=====Задания====='''
print('\nЗадание 1')
ans_1 = {}
for DriverName, Rate, AutoparkName in one_to_many:
    if AutoparkName[0] == 'B': #Название автопарка начинается на B
        if AutoparkName in ans_1:
            ans_1[AutoparkName].append(DriverName)
        else:
            ans_1[AutoparkName] = [DriverName]
print(*ans_1.items())

print('\nЗадание 2')
ans_2 = {}
for TMP, Rate, AutoparkName in one_to_many:
    if AutoparkName in ans_2:
        ans_2[AutoparkName] = max(ans_2[AutoparkName], Rate)
    else:
        ans_2[AutoparkName] = Rate
print(*{key: value for key, value in sorted(ans_2.items())}.items())

print('\nЗадание 3')
ans_3 = []
for DriverName, TMP, AutoparkName in many_to_many:
    ans_3.append((AutoparkName, DriverName))
print(*sorted(ans_3, key=lambda x: len(x)))

if __name__ == '__main__':
    main()

```

Результат выполнения:

```

Задание 1
('BestTaxi', ['Samsonov']) ('Blablacar', ['Victorovich', 'Alexandrovich', 'Grigoryevich'])

Задание 2
('BestTaxi', 5.0) ('Blablacar', 4.1) ('Citymobile', 3.0)

Задание 3
('Citymobile', 'Mirshin') ('BestTaxi', 'Samsonov') ('Blablacar', 'Victorovich')

```