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)

Отчет по РК №1 по курсу
Базовые компоненты интернет-технологий

Вариант №3(Д)

ПРЕПОДАВАТЕЛЬ

_____

Гапанюк Ю. Е. (подпись)

ИСПОЛНИТЕЛЬ:

студентка группы ИУ5-35Б _____

Гурова М.Д. (подпись)

"__"_____2021 г.

Москва - 2021

Текст программы:

Файл driver.py

```
class Driver:
    def __init__(self, id, fio, sal, park_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.park_id = park_id
```

Файл park.py

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

Файл driverpark.py

```
class DriverPark:


    def __init__(self, park_id, driver_id):
        self.park_id = park_id
        self.driver_id = driver_id
```

Файл main.py

```
from driver import Driver
from park import Park
from driverpark import DriverPark
from operator import itemgetter

# Автопарки
parks = [
    Park(1, 'Автомобильный Московский Парк №1'),
    Park(2, 'Автомобильный Московский Парк №2'),
    Park(3, 'Личный водитель'),
    Park(4, 'Такси 777'),
    Park(5, 'Автопарк Москвы'),

    Park(11, '1-й московский (до обновления)'),
    Park(22, 'Автомобильный Московский Парк №2 (в прошлом №45)'),
    Park(33, 'Личный водитель (в прошлом  Таксопарк)'),
    Park(44, '1-й московский (до обновления)'),
    Park(55, 'Автопарк Москвы (в прошлом  Таксопарк)'),
```

```
]

# Сотрудники
drivers = [
    Driver(1, 'Васильев', 100000, 3),
    Driver(2, 'Лавров', 65000, 1),
    Driver(3, 'Айвазян', 70000, 2),
    Driver(4, 'Лианозов', 68000, 2),
    Driver(5, 'Григорян', 91000, 3),

    Driver(6, 'Дмитриев', 73000, 3),
    Driver(7, 'Листопадов', 66000, 4),
    Driver(8, 'Федотов', 50000, 4),
    Driver(9, 'Дорофеев', 85000, 4),
    Driver(10, 'Голубев', 90000, 5),

    Driver(11, 'Закрадзе', 90000, 5),
    Driver(12, 'Киреев', 60000, 5),
]

drivers_parks = [
    DriverPark(3,1),
    DriverPark(1,2),
    DriverPark(2,3),
    DriverPark(2,4),
    DriverPark(3,5),
    DriverPark(3,6),
    DriverPark(4,7),
    DriverPark(4,8),
    DriverPark(4,9),
    DriverPark(5,10),
    DriverPark(5,11),
    DriverPark(5,12),

    DriverPark(11,1),
    DriverPark(22,2),
    DriverPark(33,3),
    DriverPark(33,4),
    DriverPark(33,5),
```

```
    DriverPark(33,6),
    DriverPark(44,7),
    DriverPark(44,8),
    DriverPark(44,9),
    DriverPark(55,10),
    DriverPark(55,11),
    DriverPark(55,12),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(d.fio, d.sal, p.name)
        for p in parks
        for d in drivers
        if d.park_id==p.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(p.name, dp.park_id, dp.driver_id)
        for p in parks
        for dp in drivers_parks
        if p.id==dp.park_id]

    many_to_many = [(d.fio, d.sal, park_name)
        for park_name, park_id, driver_id in many_to_many_temp
        for d in drivers if d.id==driver_id]

    print('Вариант 3')
    print('Задание Д1')
    rez1 =[]
    for d in drivers:
        if "ов" in d.fio:
            p_id=d.park_id
            parkname=parks[p_id].name
            # Добавляем результат
            rez1.append((d.fio,parkname))
    print(rez1)

    print('\nЗадание Д2')
```

```
    rez2uns = []
    # Перебираем все парки
    for p in parks:
        # Список водителей парка
        p_drivers = list(filter(lambda i: i[2]==p.name, one_to_many))
        # Если в парке есть водители
        if len(p_drivers) > 0:
            # Зарплаты водителей парка
            p_sals = [sal for _,sal,_ in p_drivers]
            # Средняя зарплата водителей парка
            p_sals_sred= sum(p_sals)/len(p_drivers)
            rez2uns.append((p.name, p_sals_sred))

    # Сортировка по средней зарплате (по убыванию)
    rez2 = sorted(rez2uns, key=itemgetter(1), reverse=True)
    print(rez2)

    print('\nЗадание ДЗ')
    rez3 = {}
    # Перебираем все отделы
    for p in parks:
        if 'A' in p.name:
            # Список водителей парка
            p_drivers = list(filter(lambda i: i[2]==p.name, many_to_many))
            # Только ФИО водителей
            p_drivers_names = [x for x,_,_ in p_drivers]
            # Добавляем результат в словарь
            # ключ - парк, значение - список фамилий
            rez3[p.name] = p_drivers_names

    print(rez3)

if __name__ == '__main__':
    main()
```

```
source /Users/mac/Desktop/RKpython/rk/bin/activate
mac@MacBook-Pro-Mac RKpython % source /Users/mac/Desktop/RKpython/rk/bin/activate
(rk) mac@MacBook-Pro-Mac RKpython %  /usr/bin/env /Users/mac/Desktop/RKpython/rk/bin/python /Users/mac/.vscode/extensions/ms-python.pyt
hon-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 55088 -- /Users/mac/Desktop/RKpython/main.py
```

Результаты работы программы:

Вариант 3
Задание Д1
[('Лавров', 'Автомобильный Московский Парк №2'), ('Лианозов', 'Личный водитель'), ('Листопадо
в', 'Автопарк Москвы'), ('Федотов', 'Автопарк Москвы')]

Задание Д2
[('Личный водитель', 88000.0), ('Автопарк Москвы', 80000.0), ('Автомобильный Московский Парк
№2', 69000.0), ('Такси 777', 67000.0), ('Автомобильный Московский Парк №1', 65000.0)]

Задание Д3
{'Автомобильный Московский Парк №1': ['Лавров'], 'Автомобильный Московский Парк №2': ['Айвазя
н', 'Лианозов'], 'Автопарк Москвы': ['Голубев', 'Закрадзе', 'Киреев'], 'Автомобильный Московс
кий Парк №2 (в прошлом №45)': ['Лавров'], 'Автопарк Москвы (в прошлом Таксопарк)': ['Голубев
', 'Закрадзе', 'Киреев']}