

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

№ 1

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

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)
]

def main():
    # 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]

    # 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]

    print('Задание Д1')
    res_1 = [(i[0], i[2]) for i in one_to_many if i[0].endswith("ов")]
    print(res_1)

```

```

print('\nЗадание Д2')
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
print(res_2)

print('\nЗадание Д3')
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

print(res_3)

if __name__ == '__main__':
    main()

```

Вывод:

Задание Д1

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

Задание Д2

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

Задание Д3

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