**Отчет по РК № 1 по курсу**

**"Разработка Интернет-Приложений"**

Выполнила:

Студентка группы

ИУ5-55Б

Богданова В.В.

Москва, МГТУ – 2021

**Задание:**

1. «Водитель» и «Автопарк» связаны соотношением один-ко-многим. Выведите список всех Водителей, у которых фамилия заканчивается на «ов», и названия их Автопарков.
2. «Водитель» и «Автопарк» связаны соотношением один-ко-многим. Выведите список Автопарков со средней зарплатой Водителей в каждом Автопарке, отсортированный по средней зарплате.
3. «Водитель» и «Автопарк» связаны соотношением многие-ко-многим. Выведите список всех Автопарков, у которых название начинается с буквы «А», и список работающих в них Водителей.

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Класс 1** | **Класс 2** |
| **3** | **Водитель** | **Автопарк** |

**Текст программы:**

# используется для сортировки

from operator import itemgetter

class Driver:

    """Водитель"""

    def \_\_init\_\_(self, id, fio, sal, park\_id):

        self.id = id

        self.fio = fio

        self.sal = sal

        self.park\_id = park\_id

class CarPark:

    """Автопарк"""

    def \_\_init\_\_(self, id, name):

        self.id = id

        self.name = name

class DrivCarPark:

    """

    'Водители автопарка' для реализации

    связи многие-ко-многим

    """

    def \_\_init\_\_(self, park\_id, driv\_id):

        self.park\_id = park\_id

        self.driv\_id = driv\_id

# Автопарки

parks = [

    CarPark(1, 'Яндекс такси'),

    CarPark(2, 'Ситимобил'),

    CarPark(3, 'Автогортранс'),

    CarPark(11, 'Делимобиль'),

    CarPark(22, 'Максим'),

    CarPark(33, 'Абер'),

]

# Водители

drivs = [

    Driver(1, 'Артамонов', 25000, 1),

    Driver(2, 'Петров', 35000, 2),

    Driver(3, 'Иваненко', 45000, 3),

    Driver(4, 'Иванов', 35000, 3),

    Driver(5, 'Иванин', 25000, 3),

]

parks\_drivs = [

    DrivCarPark(1,1),

    DrivCarPark(2,2),

    DrivCarPark(3,3),

    DrivCarPark(3,4),

    DrivCarPark(3,5),

    DrivCarPark(11,1),

    DrivCarPark(22,2),

    DrivCarPark(33,1),

    DrivCarPark(33,2),

    DrivCarPark(33,5),

]

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 drivs

        if d.park\_id==p.id]

    # Соединение данных многие-ко-многим

    many\_to\_many\_temp = [(p.name, pd.park\_id, pd.driv\_id)

        for p in parks

        for pd in parks\_drivs

        if p.id==pd.park\_id]

    many\_to\_many = [(d.fio, d.sal, park\_name)

        for park\_name, park\_id, driv\_id in many\_to\_many\_temp

        for d in drivs if d.id==driv\_id]

    print('Задание Д1')

    res\_11 = [(d.fio, d.sal, p.name)

        for p in parks

        for d in drivs

        if d.park\_id==p.id and d.fio.endswith('ов')]

    print(res\_11)

    print('\nЗадание Д2')

    res\_12\_unsorted = []

    # Перебираем все автопарки

    for p in parks:

        # Список водителей автопарка

        p\_drivs = list(filter(lambda i: i[2]==p.name, one\_to\_many))

        # Если автопарк не пустой

        if len(p\_drivs) > 0:

            # Зарплаты водителей автопарка

            d\_sals = [sal for \_,sal,\_ in p\_drivs]

            # Средняя зарплата водителей автопарка

            d\_sals\_mean = sum(d\_sals) / len(d\_sals)

            res\_12\_unsorted.append((p.name, d\_sals\_mean))

    # Сортировка по средней зарплате

    res\_12 = sorted(res\_12\_unsorted, key=itemgetter(1), reverse=True)

    print(res\_12)

    print('\nЗадание Д3')

    res\_13 = {}

    # Перебираем все автопарки

    for p in parks:

        if p.name.lower().startswith('а'):

            # Список водителей автопарка

            drivs\_of\_p = list(filter(lambda i: i[2]==p.name, many\_to\_many))

            # Только ФИО сотрудников

            drivs\_names = [x for x,\_,\_ in drivs\_of\_p]

            # Добавляем результат в словарь

            # ключ - автопарк, значение - список фамилий

            res\_13[p.name] = drivs\_names

    print(res\_13)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

**Результат работы:**

