**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений»

Рубежный контроль №1

Вариант №3Б

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-52Б |  | преподаватель каф. ИУ5 |
| Барышников Михаил |  | Гапанюк Ю.Е. |
|  |  |  |

Москва, 2021 г.

**Описание задания:**

1. Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
2. Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
3. Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

**Вариант Б.**

1. «Автопарк» и «Водитель» связаны соотношением один-ко-многим. Выведите список всех связанных водителей и автопарков, отсортированный по водителям, сортировка по автопаркам произвольная.
2. «Автопарк» и «Водитель» связаны соотношением один-ко-многим. Выведите список автопарков с количеством водителей в каждом автопарке, отсортированный по количеству водителей.
3. «Автопарк» и «Водитель» связаны соотношением многие-ко-многим. Выведите список всех водителей, у которых фамилия заканчивается на «ов», и названия их автопарков.

**Текст программы**

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 DriverPark:

"""Водители автопарка"""

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

self.park\_id = park\_id

self.driver\_id = driver\_id

def main():

# Автопарки

parks = [

CarPark(1, 'Люблинский'),

CarPark(2, 'Дмитровкий'),

CarPark(3, 'Марьинский'),

CarPark(11, 'Чертановский'),

CarPark(22, 'Нахабинский'),

CarPark(33, 'Лобнинский'),

]

# Водители

drivers = [

Driver(1, 'Артамонов', 35000, 1),

Driver(2, 'Петров', 40000, 2),

Driver(3, 'Иваненко', 37500, 3),

Driver(4, 'Иванов', 30000, 3),

Driver(5, 'Иванин', 40000, 3),

]

drivers\_parks = [

DriverPark(1, 1),

DriverPark(2, 2),

DriverPark(3, 3),

DriverPark(3, 4),

DriverPark(3, 5),

DriverPark(11, 1),

DriverPark(22, 2),

DriverPark(33, 4),

DriverPark(33, 5),

DriverPark(33, 3),

]

res1 = sorted([(d.fio, p.name) for d in drivers for p in parks if d.park\_id == p.id], key=lambda x: x[0])

print(res1)

res2 = sorted({p.name: len(list(filter(lambda x: x.park\_id == p.id, drivers))) for p in parks}.items(), key=lambda x: x[1], reverse=True)

print(res2)

res3 = {d.fio: [p.name for p in parks if p.id in [drp.park\_id for drp in drivers\_parks if drp.driver\_id == d.id]] for d in drivers if str(d.fio).endswith('ов')}

print(res3)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Экранная форма с результатом выполнения программы:**

