Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий» Отчет по рубежному контролю №1

Вариант №3 (А)

# Выполнил: Проверил:

студент группы ИУ5-51Б преподаватель каф. ИУ5 Аляев Д. В. Гапанюк Ю. Е.

# Москва, 2021 г.

Задание:

1. «Автопарк» и «Водитель» связаны соотношением один-ко-многим. Выведите список всех связанных водителей и автопарков, отсортированный по отделам, сортировка по сотрудникам произвольная.
2. «Автопарк» и «Водитель» связаны соотношением один-ко-многим. Выведите список автопарков с суммарной зарплатой водителей в каждом автопарке, отсортированный по суммарной зарплате.
3. «Автопарк» и «Водитель» связаны соотношением многие-ко-многим. Выведите список всех автопарков, у которых в названии присутствует слово «авто», и список работающих в них водителей.

# Текст программы:

# используется для сортировки  
from operator import itemgetter  
  
  
class Driver:  
 *"""Водитель"""* def \_\_init\_\_(self, id, fio, sal, Autopark\_id):  
 self.id = id  
 self.fio = fio  
 self.sal = sal  
 self.Autopark\_id = Autopark\_id  
  
  
class Autopark:  
 *"""Автопарк"""* def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
  
class DriverAutopark:  
 *"""  
 'Водители автопарка' для реализации  
 связи многие-ко-многим  
 """* def \_\_init\_\_(self, Autopark\_id, Driver\_id):  
 self.Autopark\_id = Autopark\_id  
 self.Driver\_id = Driver\_id  
  
  
# Отделы  
autoparks = [  
 Autopark(1, 'Goodway'),  
 Autopark(2, 'Экспресс-авто'),  
 Autopark(3, 'Автопарк-Гарант'),  
  
 Autopark(11, 'Goodway (другой)'),  
 Autopark(22, 'Экспресс-авто (другой)'),  
 Autopark(33, 'Автопарк-Гарант (другой)'),  
]  
  
# Сотрудники  
drivers = [  
 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),  
]  
  
drivers\_autoparks = [  
 DriverAutopark(1, 1),  
 DriverAutopark(2, 2),  
 DriverAutopark(3, 3),  
 DriverAutopark(3, 4),  
 DriverAutopark(3, 5),  
  
 DriverAutopark(11, 1),  
 DriverAutopark(22, 2),  
 DriverAutopark(33, 3),  
 DriverAutopark(33, 4),  
 DriverAutopark(33, 5),  
]  
  
  
def main():  
 *"""Основная функция"""* # Соединение данных один-ко-многим  
 one\_to\_many = [(e.fio, e.sal, d.name)  
 for d in autoparks  
 for e in drivers  
 if e.Autopark\_id == d.id]  
  
 # Соединение данных многие-ко-многим  
 many\_to\_many\_tDriver = [(d.name, ed.Autopark\_id, ed.Driver\_id)  
 for d in autoparks  
 for ed in drivers\_autoparks  
 if d.id == ed.Autopark\_id]  
  
 many\_to\_many = [(e.fio, e.sal, Autopark\_name)  
 for Autopark\_name, Autopark\_id, Driver\_id in many\_to\_many\_tDriver  
 for e in drivers if e.id == Driver\_id]  
  
 print('Задание А1')  
 res\_11 = sorted(one\_to\_many, key=itemgetter(2))  
 print(res\_11)  
  
 print('\nЗадание А2')  
 res\_12\_unsorted = []  
 # Перебираем все автопарки  
 for d in autoparks:  
 # Список водителей автопарка  
 d\_drivers = list(filter(lambda i: i[2] == d.name, one\_to\_many))  
 # Если автопарк не пустой  
 if len(d\_drivers) > 0:  
 # Зарплаты водителей автопарка  
 d\_sals = [sal for \_, sal, \_ in d\_drivers]  
 # Суммарная зарплата водителей автопарка  
 d\_sals\_sum = sum(d\_sals)  
 res\_12\_unsorted.append((d.name, d\_sals\_sum))  
  
 # Сортировка по суммарной зарплате  
 res\_12 = sorted(res\_12\_unsorted, key=itemgetter(1), reverse=True)  
 print(res\_12)  
  
 print('\nЗадание А3')  
 res\_13 = {}  
 # Перебираем все автопарки  
 for d in autoparks:  
 # Ищем слово "авто" в названии автопарка вне зависимости от регистра  
 if 'авто' in d.name.lower():  
 # Список водителей автопарка  
 d\_drivers = list(filter(lambda i: i[2] == d.name, many\_to\_many))  
 # Только ФИО водителей  
 d\_drivers\_names = [x for x, \_, \_ in d\_drivers]  
 # Добавляем результат в словарь  
 # ключ - автопарк, значение - список фамилий  
 res\_13[d.name] = d\_drivers\_names  
  
 print(res\_13)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

# Результаты программы:

Задание А1

[('Артамонов', 25000, 'Goodway'), ('Иваненко', 45000, 'Автопарк-Гарант'), ('Иванов', 35000, 'Автопарк-Гарант'), ('Иванин', 25000, 'Автопарк-Гарант'), ('Петров', 35000, 'Экспресс-авто')]

Задание А2

[('Автопарк-Гарант', 105000), ('Экспресс-авто', 35000), ('Goodway', 25000)]

Задание А3

{'Экспресс-авто': ['Петров'], 'Автопарк-Гарант': ['Иваненко', 'Иванов', 'Иванин'], 'Экспресс-авто (другой)': ['Петров'], 'Автопарк-Гарант (другой)': ['Иваненко', 'Иванов', 'Иванин']}

Process finished with exit code 0