**Московский государственный технический университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по рубежному контролю №1

Вариант №20 (А)

|  |  |
| --- | --- |
| Выполнил: | Проверил: |
| студент группы ИУ5-51Б | преподаватель каф. ИУ5 |
| Попов Н. А. | Гапанюк Ю. Е. |

Москва, 2021 г.

**Задание:**

1. «Поставщик» и «Деталь» связаны соотношением один-ко-многим. Выведите список всех связанных поставщиков и деталей, отсортированный по поставщикам, сортировка по деталям произвольная.
2. «Поставщик» и «Деталь» связаны соотношением один-ко-многим. Выведите список поставщиков с суммарной стоимостью деталей, отсортированный по суммарной стоимости.
3. «Поставщик» и «Деталь» связаны соотношением многие-ко-многим. Выведите список всех поставщиков, у которых в названии присутствует слово «Motors», и список поставляемых ими деталей.

**Текст программы:**

# используется для сортировки

from operator import itemgetter

class det:

"""Деталь"""

def \_\_init\_\_(self, id, fio, price, dep\_id):

self.id = id

self.fio = fio

self.sal = price

self.dep\_id = dep\_id

class mark:

"""Поставщик"""

def \_\_init\_\_(self, id, name):

self.id = id

self.name = name

class DetMark:

def \_\_init\_\_(self, dep\_id, emp\_id):

self.dep\_id = dep\_id

self.emp\_id = emp\_id

# Отделы

mark = [

mark(1, 'Mitsubishi Motors'),

mark(2, 'Honda Motors'),

mark(3, 'Ford Motors'),

mark(11, 'Toyota Motors'),

mark(22, 'Tesla Company'),

mark(33, 'VAG'),

]

# Сотрудники

det = [

det(1, 'Crankshaft', 18200, 1),

det(2, 'Piston', 7350, 2),

det(3, 'Gear set', 45000, 1),

det(4, 'Water Pump', 12000, 3),

det(5, 'Ignition coil', 1200, 3),

]

dets\_marks = [

DetMark(1,1),

DetMark(2,2),

DetMark(3,3),

DetMark(3,4),

DetMark(3,5),

DetMark(11,1),

DetMark(22,2),

DetMark(33,3),

DetMark(33,4),

DetMark(33,5),

]

def main():

"""Основная функция"""

# Соединение данных один-ко-многим

one\_to\_many = [(d.fio, d.sal, m.name)

for m in mark

for d in det

if d.dep\_id==m.id]

# Соединение данных многие-ко-многим

many\_to\_many\_temp = [(d.name, ed.dep\_id, ed.emp\_id)

for d in mark

for ed in dets\_marks

if d.id==ed.dep\_id]

many\_to\_many = [(e.fio, e.sal, dep\_name)

for dep\_name, dep\_id, emp\_id in many\_to\_many\_temp

for e in det if e.id==emp\_id]

print('Задание А1')

res\_11 = sorted(one\_to\_many, key=itemgetter(2))

print(res\_11)

print('\nЗадание А2')

res\_12\_unsorted = []

for m in mark:

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

if len(m\_dets) > 0:

# Зарплаты сотрудников отдела

m\_prices = [sal for \_,sal,\_ in m\_dets]

# Суммарная зарплата сотрудников отдела

d\_prices\_sum = sum(m\_prices)

res\_12\_unsorted.append((m.name, d\_prices\_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 отделы

for m in mark:

if 'Motors' in m.name:

# Список сотрудников отдела

m\_dets = list(filter(lambda i: i[2]==m.name, many\_to\_many))

# Только ФИО сотрудников

m\_dets\_names = [x for x,\_,\_ in m\_dets]

res\_13[m.name] = m\_dets\_names

print(res\_13)

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

main()

**Результат выполнения программы:**

Задание А1

[('Water Pump', 12000, 'Ford Motors'), ('Ignition coil', 1200, 'Ford Motors'), ('Piston', 7350, 'Honda Motors'), ('Crankshaft', 18200, 'Mitsubishi Motors'), ('Gear set', 45000, 'Mitsubishi Motors')]

Задание А2

[('Mitsubishi Motors', 63200), ('Ford Motors', 13200), ('Honda Motors', 7350)]

Задание А3

{'Mitsubishi Motors': ['Crankshaft'], 'Honda Motors': ['Piston'], 'Ford Motors': ['Gear set', 'Water Pump', 'Ignition coil'], 'Toyota Motors': ['Crankshaft']}

Press any key to continue . . .