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

Факультет «Информатика и управление» Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений» Отчет по рубежному контролю №1

Выполнил: Проверил:

студент группы ИУ5-51Б преподаватель каф. ИУ5

Гапчук Л.Д. Гапанюк Ю.Е.

Подпись и дата: Подпись и дата:

г. Москва, 2020 г.

**Гапчук Л.Д. ИУ5-51Б Вариант А6**

1. «Улица» и «Дом» связаны соотношением один-ко-многим. Выведите список всех связанных домов и улиц, отсортированный по улицам, сортировка по домам произвольная.

2. «Улица» и «Дом» связаны соотношением один-ко-многим. Выведите список улиц с суммарной стоимостью домов на каждой улице, отсортированный по суммарной стоимости.

3. «Улица» и «Дом» связаны соотношением многие-ко-многим. Выведите список всех улиц, у которых в названии присутствует слово «Чичерина», и список домов на этих улицах.

**Текст программы**

**main.py**

# используется для сортировки   
from operator import itemgetter   
  
class House:   
"""Дом"""   
def \_\_init\_\_(self, id, fio, sal, Street\_id):   
self.id = id   
self.fio = fio   
self.sal = sal   
self.Street\_id = Street\_id   
  
class Street:   
"""Улица"""   
def \_\_init\_\_(self, id, name):   
self.id = id   
self.name = name   
  
class HouseStreet:   
"""   
'Дома на улице' для реализации   
связи многие-ко-многим   
"""   
def \_\_init\_\_(self, Street\_id, House\_id):   
self.Street\_id = Street\_id   
self.House\_id = House\_id   
  
# Улицы   
Streets = [   
Street(1, 'Ярославское шоссе'),   
Street(2, 'Улица Чичерина'),   
Street(3, 'Улица Енисейская'),   
  
]   
  
# Дома   
Houses = [   
House(1, '123', 25000, 1),   
House(2, '119к1', 35000, 2),   
House(3, '34', 45000, 3),   
House(4, '36А', 35000, 3),   
House(5, '8к2', 25000, 3),   
]   
  
Houses\_Streets = [   
HouseStreet(1,1),   
HouseStreet(2,2),   
HouseStreet(3,3),   
HouseStreet(3,4),   
HouseStreet(3,5),   
  
]   
  
  
def main():   
"""Основная функция"""   
  
# Соединение данных один-ко-многим   
one\_to\_many = [(e.fio, e.sal, d.name)   
for d in Streets   
for e in Houses   
if e.Street\_id==d.id]   
  
# Соединение данных многие-ко-многим   
many\_to\_many\_tHouse = [(d.name, ed.Street\_id, ed.House\_id)   
for d in Streets   
for ed in Houses\_Streets   
if d.id==ed.Street\_id]   
  
many\_to\_many = [(e.fio, e.sal, Street\_name)   
for Street\_name, Street\_id, House\_id in many\_to\_many\_tHouse   
for e in Houses if e.id==House\_id]   
  
print('Задание А1')   
res\_11 = sorted(one\_to\_many, key=itemgetter(2))   
print(res\_11)   
  
print('\nЗадание А2')   
res\_12\_unsorted = []   
# Перебираем все отделы   
for d in Streets:   
# Список домов   
d\_Houses = list(filter(lambda i: i[2]==d.name, one\_to\_many))   
# Если улица не пустой   
if len(d\_Houses) > 0:   
# Цены на дома   
d\_sals = [sal for \_,sal,\_ in d\_Houses]   
# Суммарная стоимость домов   
d\_sals\_sum = sum(d\_sals)   
res\_12\_unsorted.append((d.name, d\_sals\_sum))   
  
# Сортировка по суммарной стоимости домов   
res\_12 = sorted(res\_12\_unsorted, key=itemgetter(1), reverse=True)   
print(res\_12)   
  
print('\nЗадание А3')   
res\_13 = {}   
# Перебираем улицы   
for d in Streets:   
if 'Чичерина' in d.name:   
# Список улиц   
d\_Houses = list(filter(lambda i: i[2]==d.name, many\_to\_many))   
# Только дома   
d\_Houses\_names = [x for x,\_,\_ in d\_Houses]   
# Добавляем результат в словарь   
# ключ - отдел, значение - список домов   
res\_13[d.name] = d\_Houses\_names   
  
print(res\_13)   
  
if \_\_name\_\_ == '\_\_main\_\_':   
main()

**Примеры работы программы**

Задание А1   
[('34', 45000, 'Улица Енисейская'), ('36А', 35000, 'Улица Енисейская'), ('8к2', 25000, 'Улица Енисейская'), ('119к1', 35000, 'Улица Чичерина'), ('123', 25000, 'Ярославское шоссе')]   
  
Задание А2   
[('Улица Енисейская', 105000), ('Улица Чичерина', 35000), ('Ярославское шоссе', 25000)]   
  
Задание А3   
{'Улица Чичерина': ['119к1']}