

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

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Система обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»
Рубежный контроль №1
Вариант Г №20

Выполнил:

Студент ИУ5-34Б

Фролов М. К.

Подпись и Дата:

Проверил:

Преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и Дата:

Москва, 2023 г.

Задание:

Вариант Г.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

20	Деталь	Поставщик
----	--------	-----------

Реализация:

```
# Вариант Г.
# 1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите
# список всех отделов,
# у которых название начинается с буквы «А», и список работающих в них
# сотрудников.
# 2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите
# список
# отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный
# по максимальной зарплате.
# 3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите
# список всех связанных сотрудников и отделов, отсортированный по отделам,
# сортировка по сотрудникам произвольная.

# вариант 20 - деталь и поставщик

# используется для сортировки
from operator import itemgetter
from random import randint

class Detail:                                #класс-реализация деталей
    def __init__(self, id, name, price, sup_id):
        self.id = id
        self.name = name
        self.price = price
        self.sup_id = sup_id

class Suppliers:                             #класс-реализация поставщиков
    def __init__(self, id, name):
        self.id = id
        self.name = name

class EmpDep:                               # класс для реализации связи многие-ко-многим
    def __init__(self, sup_id, det_id):
        self.sup_id = sup_id
        self.det_id = det_id

suppliers = []                              # сведения о поставщиках
```

```

Suppliers(1, 'Всё для дома поставщик'),
Suppliers(2, 'Поставкиру поставщик'),
Suppliers(3, 'Аризона поставщик'),
Suppliers(11, 'Всё для дома поставщик(другой)'),
Suppliers(22, 'Поставкиру поставщик(другой)'),
Suppliers(33, 'Аризона поставщик(другая)'),
]
det_names = ["Гвоздь", "Шестерня", "Крышка", "Втулка", "Винт"]
details = [
    # сведения о деталях + случайными числами
    Detail(1, det_names[0], randint(5000, 10000), 1),
    Detail(2, det_names[1], randint(5000, 10000), 2),
    Detail(3, det_names[2], randint(5000, 10000), 3),
    Detail(4, det_names[3], randint(5000, 10000), 3),
    Detail(5, det_names[4], randint(5000, 10000), 3),
]
emps_deps = [
    EmpDep(1,1),
    EmpDep(2,2),
    EmpDep(3,3),
    EmpDep(3,4),
    EmpDep(3,5),
    EmpDep(11,1),
    EmpDep(22,2),
    EmpDep(33,3),
    EmpDep(33,4),
    EmpDep(33,5),
]
def main(): #сновная функция
    # Соединение данных один-ко-многим
    one_to_many = [(e.name, e.price, d.name) for d in suppliers for e in details
if e.sup_id==d.id]
    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, ed.sup_id, ed.det_id) for d in suppliers for ed
in emps_deps if d.id==ed.sup_id]

    many_to_many = [(e.name, e.price, sup_name) for sup_name, sup_id, det_id in
many_to_many_temp for e in details if e.id==det_id]

    print('Задание A1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    i = 0
    flags = []
    while i < len(res_11):
        if res_11[i][2][0] == "A":
            flags.append(i)
            i += 1
    if len(flags) == 0:
        print("поставщиков на А нет")
    else:
        print("поставщики на А + товары:")
        for i in flags:

```

```

        print(res_11[i])
print('\nЗадание A2')
res_12_unsorted = []
for d in suppliers:
    d_dets = list(filter(lambda i: i[2]==d.name, one_to_many))
    if len(d_dets) > 0:
        d_prices = [price for _, price, _ in d_dets]
        d_sals_sum = sum(d_prices)
        res_12_unsorted.append((d.name, d_sals_sum))
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)
print('\nЗадание A3')
res_13 = {}
for d in suppliers:
    if 'поставщик' in d.name:
        d_emps = list(filter(lambda i: i[2]==d.name, many_to_many))
        d_emps_names = [x for x, _, _ in d_emps]
        res_13[d.name] = d_emps_names
print(res_13)
if __name__ == '__main__':
    main()

```

Результат выполнения программы:

```

Задание A1
поставщики на А + товары:
('Крышка', 7770, 'Аризона поставщик')
('Втулка', 6725, 'Аризона поставщик')
('Винт', 5824, 'Аризона поставщик')

Задание A2
[('Аризона поставщик', 20319), ('Всё для дома поставщик', 8140), ('Поставкиру поставщик', 5930)]

Задание A3
{'Всё для дома поставщик': ['Гвоздь'], 'Поставкиру поставщик': ['Шестерня'], 'Аризона поставщик': ['Крышка', 'Втулка', 'Винт'], 'Всё для дома поставщик (другой)': ['Гвоздь'], 'Поставкиру поставщик(другой)': ['Шестерня'], 'Аризона поставщик(другая)': ['Крышка', 'Втулка', 'Винт']}
(VirtualEnv) PS C:\Users\flmmf2\.venv\Scripts>

```