

**Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ	«Информатики и систем управления»
КАФЕДРА	Системы обработки информации и управления

Дисциплина «Разработка Интернет-Приложений»

РУБЕЖНЫЙ КОНТРОЛЬ №1

Вариант 21

Студент	Рабцевич К. Р. ИУ5-52Б
Преподаватель	Гапанюк Ю. Е.

1. Задание

№ варианта	Класс 1	Класс 2
21	Оператор	Язык программирования

Вариант Б.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим.
Выведите список всех связанных сотрудников и отделов, отсортированный по сотрудникам, сортировка по отделам произвольная.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим.
Выведите список отделов с количеством сотрудников в каждом отделе, отсортированный по количеству сотрудников.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим.
Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.

2. Текст программы

```
from operator import itemgetter

class Oper:
    #Оператор
    def __init__(self, id, name, num, prl_id):
        self.id = id
        self.name = name
        self.num = num
        assert isinstance(prl_id, object)
        self.prl_id = prl_id

class PrLg:
    #Языки программирования
    def __init__(self, id, name):
        self.id = id
        self.name = name

class OpProg:
    #'Операторы языка программирования'
    # для реализации связи многие-ко-многим
    def __init__(self, prlg_id, op_id):
        self.prlg_id = prlg_id
        self.op_id = op_id
```

```

# Языки программирования
prlgs = [
    PrLg(1, 'Pascal'),
    PrLg(2, 'C++'),
    PrLg(3, 'Python'),
    PrLg(4, 'Java'),
    PrLg(5, 'Go'),
    PrLg(6, 'C#'),
]

# Операторы
ops = [
    Oper(1, '+ сложение', 25, 1),
    Oper(2, '- вычитание', 35, 2),
    Oper(3, '* умножение', 45, 3),
    Oper(4, '/ деление', 35, 3),
    Oper(5, '** возведение в степень', 25, 3),
]

opprogs = [
    OpProg(1, 1),
    OpProg(1, 2),
    OpProg(1, 3),
    OpProg(1, 4),
    OpProg(2, 1),
    OpProg(2, 2),
    OpProg(2, 3),
    OpProg(2, 4),
    OpProg(3, 5),
    OpProg(3, 1),
    OpProg(3, 2),
    OpProg(3, 3),
    OpProg(3, 4),
    OpProg(4, 1),
    OpProg(5, 2),
    OpProg(6, 2),
    OpProg(6, 1),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(o.name, o.num, p.name)
                    for p in prlgs
                    for o in ops
                    if o.prl_id == p.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(p.name, op.prlg_id, op.op_id)
                           for p in prlgs
                           for op in opprogs
                           if p.id == op.prlg_id]

    many_to_many = [(o.name, o.num, pl_name)
                     for pl_name, prlg_id, op_id in many_to_many_temp
                     for o in ops if o.id == op_id]

```

```

print('Задание Б1')
res_11 = sorted(one_to_many, key=itemgetter(0))
print(*res_11, sep='\n')

print('\nЗадание Б2')
res_12_unsorted = []
# Перебираем все языки программирования
for p in prlgs:
    # Список операторов ЯП
    p_ops = list(filter(lambda i: i[2] == p.name, one_to_many))
    res_12_unsorted.append((p.name, len(p_ops)))

# Сортировка по операторам
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\nЗадание Б3')
res_13 = {}
# Перебираем все ЯП
for p in ops:
    if p.name[-5] == 'ж':
        # Список операторов
        p_ops = list(filter(lambda i: i[0] == p.name, many_to_many))
        # Только названия ЯП
        p_ops_names = [x for _, x in p_ops]
        # Добавляем результат в словарь
        # ключ - ЯП, значение - список операторов
        res_13[p.name] = p_ops_names

print(res_13)

if __name__ == '__main__':
    main()

```

3. Результат выполнения программы

Задание Б1

```

(* умножение', 45, 'Python')
(** возведение в степень', 25, 'Python')
(+ сложение', 25, 'Pascal')
(- вычитание', 35, 'C++')
(/ деление', 35, 'Python')

```

Задание Б2

```

[('Python', 3), ('Pascal', 1), ('C++', 1), ('Java', 0), ('Go', 0), ('C#', 0)]

```

Задание Б3

```

{'+ сложение': ['Pascal', 'C++', 'Python', 'Java', 'C#'], '* умножение': ['Pascal', 'C++', 'Python']}

```