

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

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

Курс «Разработка интернет приложений»

Рубежный контроль №1

Вариант №1Б

Выполнил:

студент группы ИУ5-52Б  
Алексеев А. С.

Проверил:

преподаватель каф. ИУ5  
Гапанюк Ю.Е.

Москва, 2021 г.

## Условия

- 1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
- 2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
- 3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

## Код программы

```
# используется для сортировки
from operator import itemgetter

class Stud:
    """Студент"""

    def __init__(self, id, fio, averper, gro_id):
        self.id = id
        self.fio = fio
        self.averper = averper
        self.gro_id = gro_id

class Gro:
    """Группа"""

    def __init__(self, id, name):
        self.id = id
        self.name = name

class StudGro:
    """
    'Студент группы' для реализации
    связи многие-ко-многим
    """

    def __init__(self, gro_id, stud_id):
        self.gro_id = gro_id
        self.stud_id = stud_id
```

```

# Группы
gros = [
    Gro(1, 'ИУ5 - 51Б'),
    Gro(2, 'ИУ6 - 52Б'),
    Gro(3, 'ИУ7 - 53Б'),

    Gro(11, 'ИУ5'),
    Gro(22, 'ИУ6'),
    Gro(33, 'ИУ7'),
]

# Студенты
studs = [
    Stud(1, 'Артамонов', 3, 1),
    Stud(2, 'Петров', 4, 2),
    Stud(3, 'Иваненко', 5, 3),
    Stud(4, 'Иванов', 4, 3),
    Stud(5, 'Иванин', 3, 3),
]

studs_gros = [
    StudGro(1, 1),
    StudGro(2, 2),
    StudGro(3, 3),
    StudGro(3, 4),
    StudGro(3, 5),

    StudGro(11, 1),
    StudGro(22, 2),
    StudGro(33, 3),
    StudGro(33, 4),
    StudGro(33, 5),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(s.fio, s.averper, g.name)
                    for g in gros
                    for s in studs
                    if s.gro_id == g.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(g.name, sg.gro_id, sg.stud_id)
                           for g in gros
                           for sg in studs_gros
                           if g.id == sg.gro_id]

```

```

many_to_many = [(s.fio, s.averper, gro_id)
                 for gro_name, gro_id, stud_id in many_to_many_temp
                 for s in studs if s.id == stud_id]

print('Задание A1')
res_11 = sorted(one_to_many, key=itemgetter(2))
print(res_11)

print('\nЗадание A2')
res_12_unsorted = []
# Перебираем все группы
for g in gros:
    # Список студентов группы
    g_studs = list(filter(lambda i: i[2] == g.name, one_to_many))
    # Если группа не пустая
    if len(g_studs) > 0:
        # Все средние оценки студентов группы
        g_averpers = [averper for _, averper, _ in g_studs]
        # Средняя оценка группы
        g_averper_gro = sum(g_averpers)/len(g_studs)
        res_12_unsorted.append((g.name, g_averper_gro))

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

print('\nЗадание A3')
res_13 = { }
# Перебираем все группы
for g in gros:
    if 'ИУ5' in g.name:
        # Список студентов группы
        g_studs = list(filter(lambda i: i[2] == g.name, many_to_many))
        # Только ФИО студентов
        g_studs_names = [x for x, _, _ in g_studs]
        # Добавляем результат в словарь
        # ключ - группа, значение - список фамилий
        res_13[g.name] = g_studs_names

print(res_13)
wait = input("PRESS ENTER TO CONTINUE.")

if __name__ == '__main__':
    main()

```

## Результаты программы

```

Задание A1
[('Артамонов', 3, 'ИУ5 - 51Б'), ('Петров', 4, 'ИУ6 - 52Б'), ('Иваненко', 5, 'ИУ7 - 53Б'), ('Иванов', 4, 'ИУ7 - 53Б'), ('Иванин', 3, 'ИУ7 - 53Б')]

Задание A2
[('ИУ6 - 52Б', 4.0), ('ИУ7 - 53Б', 4.0), ('ИУ5 - 51Б', 3.0)]

Задание A3
{'ИУ5 - 51Б': [], 'ИУ5': []}

```