



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

ФАКУЛЬТЕТ

«Радиотехнический»

КАФЕДРА

ИУ-5 «Системы обработки информации и управления»

**Отчет по рубежному контролю № 1 по
курсу Разработка интернет-приложений
11 вариант**

Выполнил: Пшенин И. В.
Группа: РТ5-51Б

Дата
выполнения: «23» октября 2020 г.

Подпись: _____

Проверил: Гапанюк Ю. Е.

Дата
проверки: «23» октября 2020 г.

Подпись: _____

Москва, 2020 г.

Задание на рубежный контроль

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).

Для реализации запроса №2 введите в класс, находящийся на стороне связи «много», произвольный количественный признак, например, «зарплата сотрудника».

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате. Средняя зарплата должна быть округлена до 2 знака после запятой (*отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений; для округления необходимо использовать функцию <https://docs.python.org/3/library/functions.html#round>*).
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов.

№ варианта	Класс 1	Класс 2
17	Дирижер	Оркестр

Решение рубежного контроля

```
# используется для сортировки
from operator import itemgetter

class Orch:
    # Оркестр
    def __init__(self, id, title, quantity, rating, dirizh_id):
        self.id = id
        self.title = title
        self.quantity = quantity
        self.rating = rating
        self.dirizh_id = dirizh_id

class Dirizh:
    # Дирижер
    def __init__(self, id, name):
        self.id = id
        self.name = name

class OrchDir:
    # 'Дирижеров оркестра' для реализации
    # связи многие-ко-многим

    def __init__(self, dirizh_id, orch_id):
        self.dirizh_id = dirizh_id
        self.orch_id = orch_id

# Дирижеры
dirs = [
    Dirizh(1, 'Булков'),
    Dirizh(2, 'Васильев'),
    Dirizh(3, 'Арсеньев'),

    Dirizh(11, 'Булков (ученик)'),
    Dirizh(22, 'Васильев (ученик)'),
    Dirizh(33, 'Арсеньев (ученик)'),
]

# Оркестры
orchs = [
    Orch(1, 'Московский', 40, 4, 1),
    Orch(2, 'Тульский', 35, 3, 2),
    Orch(3, 'ЛунныйНЗ', 48, 5, 3),
    Orch(4, 'МарсианскийНЗ', 55, 4, 3),
    Orch(5, 'НептунскийНЗ', 50, 3, 3),
]

# НЗ = не земной \ (0.0)/
```

```

dirs_orchs = [
    OrchDir(1,1),
    OrchDir(2,2),
    OrchDir(3,3),
    OrchDir(3,4),
    OrchDir(3,5),

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

def main():
    # Основная функция

    # Соединение данных один-ко-многим
    one_to_many = [(o.title, o.quantity, d.name)
                    for d in dirs
                    for o in orchs
                    if o.dirizh_id == d.id]

#Решение задания E1
#Выберем те оркестры, у которых в названии есть "НЗ" и выведем их наименования
#и имена дирижеров

    print('\nЗадание E1\n')
    res_1 = ''
    for i in one_to_many:
        if "НЗ" in i[0]:
            res_1 = res_1 + str(i[0]) + ' ' + str(i[1]) + ' ' + str(i[2]) + '\n'
    print(res_1)

#Решение задания E2
#Выберем для каждого оркестра среднее количество участников и
#выведем эти данные предварительно отсортировав и сопоставив с дирижерами

    print("Задание E2\n")
    d_orch_all = list()
    for d in dirs:
        # Выберем все оркестры под управлением дирижера
        orchList = list(filter(lambda x: d.name == x[2], one_to_many))
        d_orch = 0
        if len(orchList) > 0:
            # Рассматривая каждый элемент списка всех оркестров
            for item in orchList:
                # Выбор значения количества файлов программы
                o = item[1]

```

```

        d_orch = d_orch + o
        # Находим среднее значение размеров файлов
        d_orch = round(d_orch/len(orchList),1)
        # Добавляем найденное среднее значение в список для вывода данных
        d_orch_all.append((d.name, d_orch))
res_2 = sorted(d_orch_all, key=itemgetter(1), reverse = True)
print(res_2)

# Соединение данных многие-ко-многим
many_to_many_temp = [(d.name, do.dirizh_id, do.orch_id)
    for d in dirs
    for do in dirs_orchs
    if d.id == do.dirizh_id]

many_to_many = [(o.title, o.quantity, dirizh_name)
    for dirizh_name, dirizh_id, orch_id in many_to_many_temp
    for o in orchs if o.id == orch_id]

#Решение задания E3
#Выберем данные из составленных связей многие-ко-многим, рассмотрим те
#оркестры название которых начинается на "M"

print("\nЗадание E3\n")
res_3 = ''
#     res_3_temp = list()
#     for o in orchs:
#         if 'M' in o.title[0]:
#             res_3_temp.append((o.title, o.quantity))

for i in many_to_many:
    str3 = i[0]
    for k in range(len(str3)):
        if k == 0 and str3[k] == 'M':
            res_3 = res_3 + str3 + ' ' + str(i[1]) + ' ' + str(i[2]) + '\n'
            break
        else:
            break
print(res_3)

if __name__ == '__main__':
    main()

```

Экранная форма с выполнением программы

Задание E1

ЛунныйИЗ 48 Арсеньев
МарсианскийИЗ 55 Арсеньев
НептунскийИЗ 50 Арсеньев

Задание E2

[('Арсеньев', 51.0), ('Булков', 40.0), ('Васильев', 35.0)]

Задание E3

Московский 40 Булков
МарсианскийИЗ 55 Арсеньев
Московский 40 Булков (ученик)
МарсианскийИЗ 55 Арсеньев (ученик)