

Текст программы

```
# ВАРИАНТ Е :
# 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.
# 3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех
# сотрудников,
# у которых фамилия начинается с буквы «А», и названия их отделов.
# ВАРИАНТ ПРЕДМЕТНОЙ ОБЛАСТИ № 2 :
# Школьник и Класс

from operator import itemgetter

class Student:
    """Школьник"""
    def __init__(self, student_id, surname, gpa, class_id):
        self.student_id = student_id
        self.surname = surname
        self.gpa = gpa
        self.class_id = class_id

class Class:
    """Класс"""
    def __init__(self, class_id, name):
        self.class_id = class_id
        self.name = name

class ClassStudents:
    """
    'Ученики класса' для реализации связи многие-ко-многим
    """
    def __init__(self, class_id, student_id):
        self.class_id = class_id
        self.student_id = student_id

# Классы
classes = [
    Class(1, '9 мат'),
    Class(2, '10 мат'),
    Class(3, '11 мат'),

    Class(11, '9 гым'),
    Class(22, '10 гым'),
    Class(33, '11 гым'),
]

# Ученики
```

```

students = [
    Student(1, 'Алексеев', 4.0, 1),
    Student(2, 'Борисов', 4.8, 2),
    Student(3, 'Иванов', 5.0, 3),
    Student(4, 'Петров', 4.4, 3),
    Student(5, 'Сидоров', 3.9, 3),
    Student(6, 'Антонов', 4.0, 3),
]

class_students = [
    ClassStudents(1, 1),
    ClassStudents(2, 2),
    ClassStudents(3, 3),
    ClassStudents(3, 4),
    ClassStudents(3, 5),
    ClassStudents(3, 6),

    ClassStudents(11, 1),
    ClassStudents(22, 2),
    ClassStudents(33, 3),
    ClassStudents(33, 4),
    ClassStudents(33, 5),
    ClassStudents(33, 6),
]

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

    one_to_many = [(s.surname, s.gpa, c.name)
                   for c in classes
                   for s in students
                   if s.class_id == c.class_id]

    many_to_many_temp = [(c.name, cs.student_id, cs.class_id)
                          for c in classes
                          for cs in class_students
                          if c.class_id == cs.class_id]
    many_to_many = [(s.surname, s.gpa, class_name)
                    for class_name, student_id, class_id in many_to_many_temp
                    for s in students if s.student_id == student_id]
    print('Задание E1')
    res_1 = {}
    # Перебираем все классы для поиска классов, содержащих 'мат'
    for c in classes:
        if 'мат' in c.name:
            c_students = list(filter(lambda i: i[2] == c.name, one_to_many)) # список учеников класса
            c_students_names = [x for x, _, _ in c_students] # только фамилии студентов
            res_1[c.name] = c_students_names
    print(res_1)

    print('\nЗадание E2')

```

```

res_2_unsorted = []
for c in classes:
    c_students = list(filter(lambda i: i[2] == c.name, one_to_many)) # список учеников класса
    if len(c_students) > 0: # если класс не пустой
        c_gpa = [gpa for _, gpa, _ in c_students]
        c_gpa_sum = round(sum(c_gpa) / len(c_students), 2)
        res_2_unsorted.append((c.name, c_gpa_sum))
res_2 = sorted(res_2_unsorted, key = itemgetter(1), reverse=True)
print(res_2)

print('\nЗадание E3')
res_3 = {}
for s in students:
    if s.surname.startswith('A'):
        s_students = list(filter(lambda i: i[0] == s.surname, many_to_many))
        s_students_surnames = [x for _, x in s_students]
        res_3[s.surname] = s_students_surnames
print(res_3)

if __name__ == '__main__':
    main()

```

Результаты вывода

Задание E1

{'9 мат': ['Алексеев'], '10 мат': ['Борисов'], '11 мат': ['Иванов', 'Петров', 'Сидоров', 'Антонов']}

Задание E2

[('10 мат', 4.8), ('11 мат', 4.33), ('9 мат', 4.0)]

Задание E3

{'Алексеев': ['9 мат', '9 гум'], 'Антонов': ['11 мат', '11 гум']}