Текст программы

# ВАРИАНТ E :

# 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 гум'),

     Class(22, '10 гум'),

     Class(33, '11 гум'),

]

# 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 Е1')

     res\_1 = {}

    # Перебираем все класы 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('А'):

               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 Е1

{'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']}