```
# ВАРИАНТ Е
# ВАРИАНТ ПРЕДМЕТНОЙ ОБЛАСТИ № 1 :
# Студент и группа
from operator import itemgetter
class Student:
"""Студент"""
def __init__(self, student_id, surname, avg_mark, group_id):
self.student_id = student_id
self.surname = surname
self.avg_mark = avg_mark
self.group_id = group_id
class Group:
"""Группа"""
def __init__(self, group_id, name):
self.group_id = group_id
self.name = name
class GroupStudents:
"""Студенты группы' для реализации связи многие-ко-многим
"""
def __init__(self, group_id, student_id):
self.group_id = group_id
self.student_id = student_id
# Группы
groups = [
Group(1,'PT5-31'),
Group(2, 'PT4-31'),
Group(3, 'PT3-31'),
```

```
Group(11,'РТ2-31'),
Group(22, 'РТ1-31'),
Group(33, 'ИУ5-31'),
]
# Студенты
students = [
Student(1, 'Ахметзянов', 4.0, 1),
Student(2, 'Бабасанова', 4.8, 2),
Student(3, 'Сидорин', 5.0, 3),
Student(4, 'Покровский', 4.4, 3),
Student(5, 'Гордин', 3.9, 3),
Student(6, 'Пересыпко', 4.0, 5),
]
group_students = [
GroupStudents(1, 1),
GroupStudents(2, 2),
GroupStudents(3, 3),
GroupStudents(3, 4),
GroupStudents(3, 5),
GroupStudents(3, 6),
GroupStudents(11, 1),
GroupStudents(22, 2),
GroupStudents(33, 3),
GroupStudents(33, 4),
GroupStudents(33, 5),
GroupStudents(33, 6),
]
```

```
def main():
"""Основная функция"""
one_to_many = [(s.surname, s.avg_mark, c.name)
for c in groups
for s in students
if s.group_id == c.group_id]
many_to_many_temp =[(c.name, cs.student_id, cs.group_id)
for c in groups
for cs in group_students
if c.group_id==cs.group_id]
many_to_many = [(s.surname, s.avg_mark, group_name)
for group_name, student_id, group_id in many_to_many_temp
for s in students if s.student_id==student_id]
print('Задание E1')
res_1 = {}
# Перебираем все группы для поиска классов, содержащих 'ИУ'
for c in groups:
if 'ИУ' in c.name:
c_students = list(filter(lambda i:i[2]==c.name, one_to_many)) # список студентов группы
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 студентов группы
```

```
if len(c_students) > 0: # если группа непустая
c_avg_mark = [avg_mark for _,avg_mark,_ in c_students]
c_avg_mark_sum = round(sum(c_avg_mark)/len(c_students), 2)
res_2_unsorted.append((c.name, c_avg_mark_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 {'ИУ5-31': ['Гордин']}

Задание E2 [('РТ5-31', 4.0), ('РТ4-31', 4.8), ('РТ3-31', 4.43), (‘РТ1-31’, 4.0]

Задание E3 {'Ахметзянов': [РТ5-31Б]}