Вариант 26Б  
**Текст программы:**

# Класс сущности "Студенческая группа"  
class Group:  
  
 def \_\_init\_\_(self, id, name, number\_of\_students):  
 self.id = id # id группы  
 self.name = name # Название группы  
 self.number\_of\_students = number\_of\_students # Кол-во студентов  
  
  
# Класс сущности "Учебный курс"  
class Course:  
  
 def \_\_init\_\_(self, id, name):  
 self.id = id # id курса  
 self.name = name # Название курса  
  
  
# Класс сущности "Студенческая группа - Учебный курс" (Для связи многие ко многим)  
class GroupCourse:  
  
 def \_\_init\_\_(self, course\_id, group\_id):  
 self.course\_id = course\_id # id курса  
 self.group\_id = group\_id # id группы  
  
  
# Cтуденческие группы  
groups = [  
 Group(1, 'IU1-12A', 26),  
 Group(2, 'IU1-11B', 28),  
 Group(3, 'IU3-21C', 20),  
 Group(4, 'IU4-42D', 24),  
 Group(5, 'IU5-44B', 19),  
]  
  
# Учебные курсы  
courses = [  
 Course(1, 'First'),  
 Course(2, 'Second'),  
 Course(3, 'Third'),  
 Course(4, 'Fourth'),  
]  
  
# Студенческая группа - Учебный курс  
groups\_courses = [  
 GroupCourse(1, 1),  
 GroupCourse(1, 2),  
 GroupCourse(2, 3),  
 GroupCourse(4, 4),  
 GroupCourse(4, 5),  
 GroupCourse(3, 5),  
]  
  
  
def main():  
 # Связь один-ко-многим  
 *"""  
 Генерируем связи один к многим путем равнозначимой  
 привязки id курса к id группы,  
 плюс одна связь пятой группы (IU5-44B) с четвертым курсом  
 """* one\_to\_many = [(g.name, g.number\_of\_students, c.name)  
 for c in courses  
 for g in groups  
 if g.id == c.id]  
 one\_to\_many.append((groups[5 - 1].name, groups[5 - 1].number\_of\_students, courses[4 - 1].name))  
  
 print('Запрос Б1')  
 b1 = sorted(one\_to\_many, key=lambda x: x[0])  
 print(b1)  
  
 print('\nЗапрос Б2')  
 b2 = []  
 for cource in courses:  
 cource\_groups = list(filter(lambda i: i[2] == cource.name, one\_to\_many))  
 if len(cource\_groups) > 0:  
 cource\_students = [students for \_, students, \_ in cource\_groups]  
 cource\_students\_sum = sum(cource\_students)  
 b2.append((cource.name, cource\_students\_sum))  
  
 b2 = sorted(b2, key=lambda x: x[1], reverse=True)  
 print(b2)  
  
 """  
 Пояснение к запросу Б3:  
 Вместо окончания "ов" в фамилии сотрудников,   
 я взял окончание "B" в имени группы  
 """  
 print('\nЗапрос Б3')  
 # Связь многие-ко-многим  
 many\_to\_many\_group = [(c.name, gc.course\_id, gc.group\_id)  
 for c in courses  
 for gc in groups\_courses  
 if c.id == gc.course\_id]  
  
 many\_to\_many = [(g.name, g.number\_of\_students, name)  
 for name, \_, group\_id in many\_to\_many\_group  
 for g in groups if g.id == group\_id]  
  
 b3 = {}  
 for g in groups:  
 if 'B' in g.name:  
 course\_groups = list(filter(lambda i: i[0] == g.name, many\_to\_many))  
 course\_groups\_names = [x for \_, \_, x in course\_groups]  
 b3[g.name] = course\_groups\_names  
  
 print(b3)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

**Результат выполнения:**Запрос Б1

[('IU1-11B', 28, 'Second'), ('IU1-12A', 26, 'First'), ('IU3-21C', 20, 'Third'), ('IU4-42D', 24, 'Fourth'), ('IU5-44B', 19, 'Fourth')]

Запрос Б2

[('Fourth', 43), ('Second', 28), ('First', 26), ('Third', 20)]

Запрос Б3

{'IU1-11B': ['First'], 'IU5-44B': ['Third', 'Fourth']}