№ варианта	Класс 1	Класс 2
7	Микропроцессор	Компьютер

Пример классов данных для предметной области Микропроцессор-Компьютер:

1. Класс «Микропроцессор», содержащий поля:
   - ID записи о Микропроцессоре;
   - Название Микропроцессора;
   - Стоимость (количественный признак);
   - ID записи о компьютере. (для реализации связи один-ко-многим)
2. Класс «Компьютер», содержащий поля:
   - ID записи о компьютере;
   - Наименование компьютера.
3. (Для реализации связи многие-ко-многим) Класс «Микропроцессор для компуктера», содержащий поля:
   - ID записи о Микропроцессоре; • ID записи о компьютере.

Вариант А.

1. «Компьютер» и «Микропроцессор» связаны соотношением один-ко-многим. Выведите список всех связанных Микропроцессоров и компьютеров, отсортированный по компьютерам, сортировка по Микропроцессорам произвольная.
2. «Компьютер» и «Микропроцессор» связаны соотношением один-ко-многим. Выведите список компьютеров с суммарной стоимостью Микропроцессоров на каждом компьютере, отсортированный по суммарной стоимости.
3. «Компьютер» и «Микропроцессор» связаны соотношением многие-комногим. Выведите список всех компьютеров, у которых в названии присутствует слово «компьютер», и список установленных на них Микропроцессоров.

Код:

```
# Гоголь И.В. ИУ5-51Б Вариант №7
# используется для сортировки
from operator import itemgetter


class Emp:
    """Микропроцессор"""

    def __init__(self, id, name, price, comp_id):
        self.id = id
```

```
        self.name = name
        self.price = price
        self.comp_id = comp_id


class Dep:
    """Компьютер"""

    def __init__(self, id, name):
        self.id = id
        self.name = name


class EmpDep:
    """
    'Микропроцессоры компьютера' для реализации
    связи многие-ко-многим
    """

    def __init__(self, comp_id, prog_id):
        self.comp_id = comp_id
        self.prog_id = prog_id


# Компьютеры
deps = [
    Dep(1, 'Компьютер Михаил'),
    Dep(2, 'Аркон'),
    Dep(3, 'Мой компьютер'),
    Dep(4, 'Леново'),
    Dep(5, 'МакБук'),
    Dep(6, 'Персональный компьютер'),
    Dep(11, 'Рабочий компьютер'),
    Dep(12, 'Соник'),
]

# Микропроцессоры
emps = [
    Emp(1, 'Pentium 4', 2500, 1),
    Emp(2, 'Intel Core I5', 3500, 2),
    Emp(3, 'Intel Core I7', 22000, 3),
    Emp(4, 'Intel Core I3', 1000, 3),
    Emp(5, 'A1', 8000, 3),
    Emp(6, 'Xeon', 500, 4),
    Emp(7, 'AMD EPYC', 800, 6),
    Emp(8, 'AMD RYZEN', 200, 6),
    Emp(9, 'Байкал', 25500, 5),
    Emp(10, 'Эльбрус 8C', 5010, 3),
    Emp(11, 'Intel Core I9', 1100, 3),
]

emps_deps = [
    EmpDep(1, 1),
    EmpDep(2, 2),
    EmpDep(3, 3),
    EmpDep(3, 4),
    EmpDep(3, 5),
    EmpDep(4, 6),
    EmpDep(6, 7),
    EmpDep(6, 8),
    EmpDep(5, 9),
    EmpDep(3, 10),
    EmpDep(3, 11),
    EmpDep(11, 1),
```

```
    EmpDep(12, 2),

]


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

    # Соединение данных один-ко-многим
    one_to_many = [(e.name, e.price, d.name)
                   for d in deps
                   for e in emps
                   if e.comp_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, ed.comp_id, ed.prog_id)
                         for d in deps
                         for ed in emps_deps
                         if d.id == ed.comp_id]

    many_to_many = [(e.name, e.price, dep_name)
                    for dep_name, dep_id, emp_id in many_to_many_temp
                    for e in emps if e.id == emp_id]

    print('Задание А1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res 11)

    print('\nЗадание А2')
    res_12_unsorted = []
    # Перебираем все компьютеры
    for d in deps:
        # Список программ компьютера
        d_emps = list(filter(lambda i: i[2] == d.name, one_to_many))
        # Если компьютер не пустой
        if len(d_emps) > 0:
            # Цена микропроцессора компьютера
            d_sals = [sal for _, sal, _ in d_emps]
            # Суммарная цена микропроцессоров компьютера
            d_sals_sum = sum(d_sals)
            res_12_unsorted.append((d.name, d_sals_sum))

    # Сортировка по суммарной цене
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)

    print('\nЗадание А3')
    res_13 = {}
    # Перебираем все компьютеры
    for d in deps:
        if 'компьютер' in d.name:
            # Список микропроцессорров компьютера
            d_emps = list(filter(lambda i: i[2] == d.name, many_to_many))
            # Только Названия микропроцессоров
            d_emps_names = [x for x, _, _ in d_emps]
            # Добавляем результат в словарь
            # ключ - компьютер, значение - список Названий
            res_13[d.name] = d_emps_names

    print(res_13)


if __name__ == '__main__':
    main()
```

Результат работы программы:

Задание A1

[('Intel Core I5', 3500, 'Аркон'), ('Pentium 4', 2500, 'Компьютер Михаил'), ('Xeon', 500, 'Леново'), ('Байкал', 25500, 'МакБук'), ('Intel Core I7', 22000, 'Мой компьютер'), ('Intel Core I3', 1000, 'Мой компьютер'), ('A1', 8000, 'Мой компьютер'), ('Эльбрус 8C', 5010, 'Мой компьютер'), ('Intel Core I9', 1100, 'Мой компьютер'), ('AMD EPYC', 800, 'Персональный компьютер'), ('AMD RYZEN', 200, 'Персональный компьютер')]

Задание A2

[('Мой компьютер', 37110), ('МакБук', 25500), ('Аркон', 3500), ('Компьютер Михаил', 2500), ('Персональный компьютер', 1000), ('Леново', 500)]

Задание A3

{'Мой компьютер': ['Intel Core I7', 'Intel Core I3', 'A1', 'Эльбрус 8C', 'Intel Core I9'], 'Персональный компьютер': ['AMD EPYC', 'AMD RYZEN'], 'Рабочий компьютер': ['Pentium 4']}

Process finished with exit code 0