Вайгент Мария
ИУ5-35Б

РК№1. Вариант 4Д

```
from numpy import *

class Computer:
    def __init__(self, id, processor_name, manufacturer, OS, disp_id):
        self.id = id
        self.processor_name = processor_name
        self.manufacturer = manufacturer
        self.OS = OS
        self.disp_id = disp_id


class Display_class:

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


class Comp_Display_class:
    """
    связь многие-ко-многим
    """

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

display_classes = [
    Display_class(1, 'Lenovo'),
    Display_class(2, 'Samsung'),
    Display_class(3, 'Huawei'),
    Display_class(4, 'Asus'),
    Display_class(5, 'Apple'),
    Display_class(6, 'DNS'),
]

computers = [
    Computer(1, 'Pentium', 'Intel', 'Windows 10', 4),
    Computer(2, 'Winchip-4', 'Centaur', 'Windows 7', 1),
    Computer(3, 'Samuel', 'VIA', 'Astra Linux', 6),
    Computer(4, 'Corvette', 'AMD', 'Windows 11', 3),
    Computer(5, 'Cayenne', 'Cyrix', 'Debian', 2),
    Computer(6, 'Apple M1', 'TSMC', 'MacOS', 5),
    Computer(7, 'Tillamook', 'Intel', 'Windows 7', 4)
]
```

```
comp_disp = [
    Comp_Display_class(4, 1),
    Comp_Display_class(1, 2),
    Comp_Display_class(6, 3),
    Comp_Display_class(3, 4),
    Comp_Display_class(2, 5),
    Comp_Display_class(5, 6),
    Comp_Display_class(4, 7)
]


def main():

    # Соединение данных один-ко-многим
    one_to_many = [(comp.processor_name, comp.manufacturer, comp.OS, disp.name)
                   for disp in display_classes
                   for comp in computers
                   if comp.disp_id == disp.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, comp_dis.display_id, comp_dis.comp_id)
                         for d in display_classes
                         for comp_dis in comp_disp
                         if d.id == comp_dis.display_id]

    many_to_many = [(comp.processor_name, comp.manufacturer, comp.OS, disp_name)
                    for disp_name, disp_id, comp_id in many_to_many_temp
                    for comp in computers if comp.id == comp_id]

# вывод компьютеров и их дисплейных классов, у которых ОС начинается на "Win"
    print('Задание Д1')

    res_11 = []
    for comp in computers:
        if comp.OS.find('Win') != -1:
            res_11.append((comp.OS, display_classes[comp.disp_id - 1].name))
    print(*res_11)

# Вывод чаще всего встречающегося производителя дисплейных класоов
    print('\nЗадание Д2')

    for d in display_classes:
        manufacturer = empty(len(display_classes))
        arr_disp = list(filter(lambda i: i[3] == d.name, one_to_many))
        manufacturer[d.id - 1] += len(arr_disp)
    max_cnt = 0
    for i in range(len(display_classes)):
        if max_cnt < manufacturer[i]:
            max_cnt = manufacturer[i]
```

```
    print(int(max_cnt))

# Вывод диспейных классов, которые начинаются с "A" и компютеров, соответствующих этим диспейным классам
    print('\nЗадание Д3')
    res_13 = {}
    for d in display_classes:
        if 'A' in d.name:
            d_emps_names = []
            arr_disp = list(filter(lambda i: i[3] == d.name, many_to_many))
            for processor_name, manufacturer, OS, _ in arr_disp:
                d_emps_names.append((processor_name, manufacturer, OS))
            res_13[d.name] = d_emps_names
    print(res_13)


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