Васильев Денис. Группа: ИУ5-52б.

Вариант 4Б

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Класс 1** | **Класс 2** |
| 4 | Компьютер | Дисплейный класс |

from operator import itemgetter

class Computer:

    def \_\_init\_\_(self, id, model, cost, class\_id):

        self.id = id

        self.model = model

        self.class\_id = class\_id

        self.cost = cost

class DisplayClass:

    def \_\_init\_\_(self, id, floor, num):

        self.id = id

        self.floor = floor

        self.num = num

class ComputerDisplayClass:

    def \_\_init\_\_(self, comp\_id, disp\_id):

        self.comp\_id = comp\_id

        self.disp\_id = disp\_id

computers = [

    Computer(1, "Asus",   40000, 1),

    Computer(2, "Dell",   30000, 1),

    Computer(3, "HP",     35000, 1),

    Computer(4, "Lenovo", 45000, 1),

    Computer(5, "Apple",  70000, 2),

    Computer(6, "Lenovo", 25000, 2),

    Computer(7, "Lenovo", 56000, 3),

    Computer(8, "Dell",   34000, 3),

]

classes = [

    DisplayClass(1, 2, 2\_33),

    DisplayClass(2, 7, 7\_18),

    DisplayClass(3, 7, 7\_20),

]

computers\_classes = [

    ComputerDisplayClass(1, 1),

    ComputerDisplayClass(1, 2),

    ComputerDisplayClass(2, 1),

    ComputerDisplayClass(2, 3),

    ComputerDisplayClass(3, 1),

    ComputerDisplayClass(4, 1),

    ComputerDisplayClass(5, 1),

    ComputerDisplayClass(6, 1),

    ComputerDisplayClass(6, 2),

    ComputerDisplayClass(6, 3),

    ComputerDisplayClass(7, 2),

    ComputerDisplayClass(8, 1),

]

def main():

    one\_to\_many = [(comp.model, comp.id, cl.floor, cl.num)

        for comp in computers

        for cl in classes

        if comp.class\_id == cl.id]

    many\_to\_many\_tmp = [(cl.floor, cl.num, cc.comp\_id, cc.disp\_id)

        for cl in classes

        for cc in computers\_classes

        if cl.id == cc.disp\_id]

    many\_to\_many = [(comp.model, comp.cost, cl\_floor, cl\_num)

        for cl\_floor, cl\_num, comp\_id, disp\_id in many\_to\_many\_tmp

        for comp in computers

        if comp.id == comp\_id]

    # Список всех связанных компьютеров и классов, отсортированный по модели компьютеров.

    print("Задание Б1")

    res1 = sorted(one\_to\_many, key=itemgetter(0))

    print(res1)

    #  Список классов с количеством компьютеров в каждом классе, отсортированный по количеству компьютеров.

    print("\nЗадание Б2")

    res2\_unsorted = []

    for cl in classes:

        cc = list(filter(lambda i: i[3]==cl.num, one\_to\_many))

        if len(cc) > 0:

            class\_comps = [comp\_id for \_,comp\_id,\_,\_ in cc]

            class\_comps\_count = len(class\_comps)

            res2\_unsorted.append((cl.num, class\_comps\_count))

    res2 = sorted(res2\_unsorted, key=itemgetter(1), reverse=True)

    print(res2)

    # Список моделей компьютеров из классов, номер которых начинается на 7 и  номера их классов

    print("\nЗадание Б3")

    res3 = {}

    for cl in classes:

        if str(cl.num).startswith('7'):

            cc = list(filter(lambda i: i[3]==cl.num, many\_to\_many))

            cc\_nums = [x for x,\_,\_,\_ in cc]

            res3[cl.num] = cc\_nums

    print(res3)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

# Результат работы программы: