Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Рубежный контроль №1

Вариант №4 (Г)

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-34Б: |  | преподаватель каф. ИУ5 |
| Даниелян Алла Армановна |  | Гапанюк Ю.В. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2022 г.

**Постановка задачи:**

**Вариант Г.**

«Отдел» и «Сотрудник» связаны соотношением один-ко-многим.

Выведите список всех отделов (Rooms) у которых название (Owner\_name) начинается с буквы «А»,

и список работающих в них (Computer.User\_name) сотрудников.

«Отдел» и «Сотрудник» связаны соотношением один-ко-многим.

Выведите список отделов с максимальной (Computer.Price) зарплатой сотрудников в каждом (Room) отделе,

отсортированный по максимальной зарплате.

«Отдел» и «Сотрудник» связаны соотношением многие-ко-многим.

Выведите список всех связанных сотрудников (Computer.User\_name)) и отделов (Room), отсортированный по отделам,

сортировка по сотрудникам произвольная.

**Выделены следующие сущности(классы):**

Компьютер (Computer)

ID\_computer

User\_Name

Price

ID\_room

Дисплейный класс (Room)

ID\_room

Owner\_name

Компьютеры дисплейного класса (Computers\_in\_room)

ID\_room

ID\_computer

**Текст программы:**

from operator import itemgetter

class Room():

    def \_\_init\_\_(self,id,on):

        self.ID\_room = id

        self.Owner\_name = on

class Computer:

    def \_\_init\_\_(self,id,usn,pr,idr):

        self.ID\_computer = id

        self.User\_name = usn

        self.Price = pr

        self.ID\_room = idr

class Computers\_in\_room:

    def \_\_init\_\_(self, cid,rid):

        self.CompID = cid

        self.RoomID = rid

room = [

    Room (1, "Иванов В.П."),

    Room (2,"Арпов А.А."),

    Room (3, "Игнатьев И.И."),

    Room (4, "Антошкин А.К."),

    Room (5, "Репкин В.В.")

]

# COMPUTERS

comp = [

    Computer (1,"Васькин", 50000,1),

    Computer (2,"Пупкин", 52000,2),

    Computer (3,"Анюткин", 60000,1),

    Computer (4,"Гринев", 30000,3),

    Computer (5,"Одинцова", 31000,4),

    Computer (10, "Десятцова", 100000,4),

    Computer (6,"Базаров", 20000,3),

    Computer (7,"Твист", 100000,5),

    Computer (8,"Дван", 760000,5),

    Computer (9,"Кран", 50000,5)

]

comp\_room = [

    Computers\_in\_room (1,1),

    Computers\_in\_room (2,2),

    Computers\_in\_room (3,3),

    Computers\_in\_room (4,4),

    Computers\_in\_room (5,5),

    Computers\_in\_room (1,6),

    Computers\_in\_room (2,7),

    Computers\_in\_room (3,8),

    Computers\_in\_room (4,9),

    Computers\_in\_room (5,1)

]

def main():

    """Основная функция"""

    # Соединение данных один-ко-многим

    one\_to\_many = [ (e.User\_name, e.Price, d.ID\_room, d.Owner\_name)

        for d in room

        for e in comp

        if e.ID\_room==d.ID\_room

        ]

    # Соединение данных многие-ко-многим

    many\_to\_many\_temp = [(d.Owner\_name, ed.CompID, ed.RoomID)

        for d in room

        for ed in comp\_room

        if d.ID\_room==ed.RoomID]

    many\_to\_many = [(e.User\_name, e.Price, own\_name, room\_id)

        for own\_name, comp\_id, room\_id in many\_to\_many\_temp

        for e in comp

        if e.ID\_computer==comp\_id ]

    print('Задание Г1')

    res = sorted(one\_to\_many, key = itemgetter(3) )

    cache = ""

    for x in res:

        if cache != x[3] and x[3][0] == 'А':

            cache = x[3]

            print("\nКомната №{}\nOтветственный: {} \nСписок пользователей:".format(x[2],x[3]))

            print (x[0])

        elif x[3][0] == 'А':

            print (x[0])

        else:

            break

    print('\nЗадание Г2')

    # Перебираем все ROOMS

    res2={}

    for d in room:

        # Список компьютеров комнаты

        room\_list = list(filter(lambda i: i[2]==d.ID\_room, one\_to\_many))

        mx = float(0)

        #Собираем списки ключей и значений

        if len(room\_list)>0:

            for r in room\_list:

              if float(r[1])>mx:

                mx = float(r[1])

            res2["Комната №{}".format(room\_list[0][2])] = mx

    #Сортируем словарь по возрастанию значений

    res\_tuples = sorted(res2.items(), key = itemgetter(1))

    print (res\_tuples)

    print('\nЗадание Г3')

    res3 = {}

    # Перебираем все ROOMS

    for d in room:

         # Список ROOMS

        rm\_comps = list(filter(lambda i: i[3]==d.ID\_room, many\_to\_many))

        # Только UserName

        rm\_comps\_unm = [x for x,\_,\_,\_ in rm\_comps]

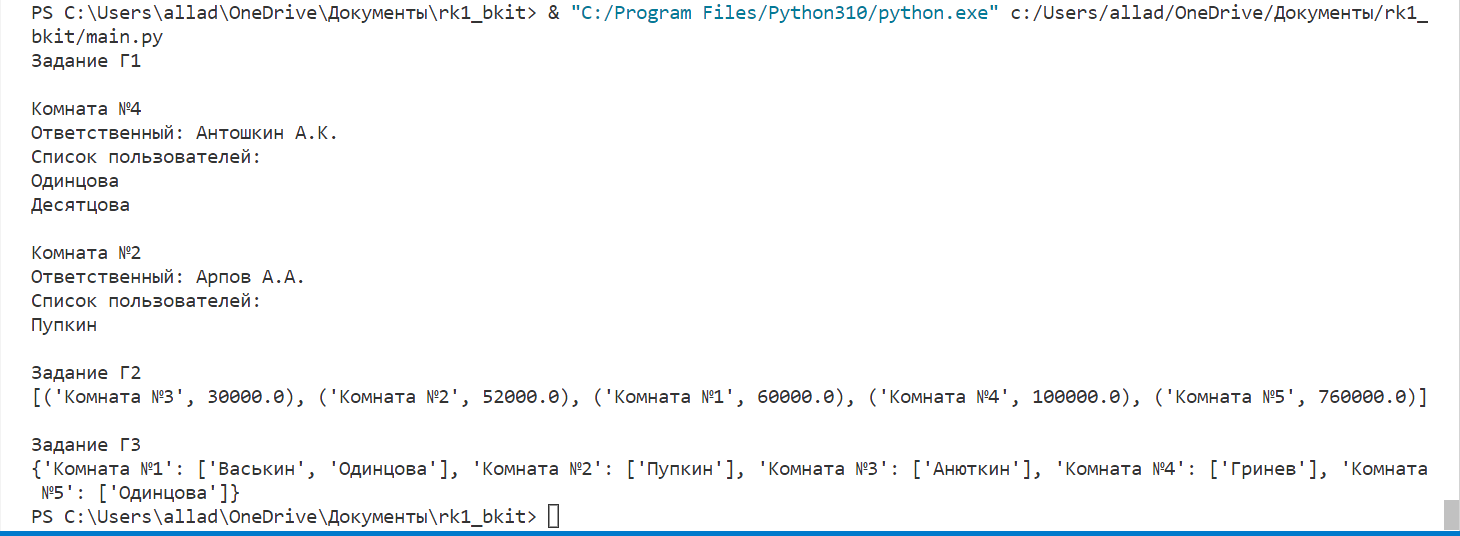
        res3["Комната №{}".format(d.ID\_room)] = rm\_comps\_unm

    print (res3)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

**Результат выполнения программы:**

****

Задание Г1

Комната №4

Oтветственный: Антошкин А.К.

Список пользователей:

Одинцова

Десятцова

Комната №2

Oтветственный: Арпов А.А.

Список пользователей:

Пупкин

Задание Г2

[('Комната №3', 30000.0), ('Комната №2', 52000.0), ('Комната №1', 60000.0), ('Комната №4', 100000.0), ('Комната №5', 760000.0)]

Задание Г3

{'Комната №1': ['Васькин', 'Одинцова'], 'Комната №2': ['Пупкин'], 'Комната №3': ['Анюткин'], 'Комната №4': ['Гринев'], 'Комната №5': ['Одинцова']}