Факультет «Информатика и системы управления»  
 Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по РК №1

Вариант Д, номер 5

Выполнил: Проверил:

студент группы ИУ5-33Б преподаватель каф. ИУ5

Гонов Марат Гапанюк Ю. Е.

Москва, 2023 г.

1. «Музыкант» и «Оркестр» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.
2. «Музыкант» и «Оркестр» связаны соотношением один-ко-многим. Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате (*отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений*).
3. «Музыкант» и «Оркестр» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.

Код программы:

class Musician:

    def \_\_init\_\_(self, id, last\_name, salary, orchestra\_id):

        self.id = id

        self.last\_name = last\_name

        self.salary = salary

        self.orchestra\_id = orchestra\_id

class Orchestra:

    def \_\_init\_\_(self, id, name):

        self.id = id

        self.name = name

class MusiciansOrchestra:

    def \_\_init\_\_(self, musician\_id, orchestra\_id):

        self.musician\_id = musician\_id

        self.orchestra\_id = orchestra\_id

musicians = [

    Musician(1, 'Smith', 30000, 1),

    Musician(2, 'Johnson', 35000, 2),

    Musician(3, 'Williams', 40000, 1),

    Musician(4, 'Jones', 32000, 2),

    Musician(5, 'Brown', 28000, 1)

]

orchestras = [

    Orchestra(1, 'Symphony Orchestra'),

    Orchestra(2, 'Chamber Orchestra')

]

musicians\_orchestras = [

    MusiciansOrchestra(1, 1),

    MusiciansOrchestra(2, 2),

    MusiciansOrchestra(3, 1),

    MusiciansOrchestra(4, 2),

    MusiciansOrchestra(5, 1)

]

# Task 1

task\_1\_result = [(m.last\_name, o.name)

                 for m in musicians

                 for o in orchestras

                 if m.orchestra\_id == o.id and m.last\_name.endswith('th')]

print("Task 1 Result:", task\_1\_result)

# Task 2

avg\_salaries = {}

for o in orchestras:

    o\_musicians = [m.salary for m in musicians if m.orchestra\_id == o.id]

    avg\_salaries[o.name] = sum(o\_musicians) / len(o\_musicians) if len(o\_musicians) > 0 else 0

sorted\_avg\_salaries = sorted(avg\_salaries.items(), key=lambda x: x[1], reverse=True)

print("Task 2 Result:", sorted\_avg\_salaries)

# Task 3

task\_3\_result = {o.name: [m.last\_name for m in musicians if m.orchestra\_id == o.id] for o in orchestras if o.name.startswith('C')}

print("Task 3 Result:", task\_3\_result)

Вывод:  
Task 1 Result: [('Smith', 'Symphony Orchestra')]

Task 2 Result: [('Chamber Orchestra', 33500.0), ('Symphony Orchestra', 32666.666666666668)]

Task 3 Result: {'Chamber Orchestra': ['Johnson', 'Jones']}