Крайников И. РТ5-31Б РК1

Вариант Е. Предметная область 17.

«Дирижер» и «Оркестр» связаны соотношением один-ко-многим. Выведите список всех дирижёров, у которых в названии присутствует слово «алекс», и список их орекстров.

«Дирижер» и «Оркестр» связаны соотношением один-ко-многим. Выведите список дирижёров со средним годом основания их орекстров, отсортированный по этой величине. Она должна быть округлена до 2 знака после запятой.

«Дирижер» и «Оркестр» связаны соотношением многие-ко-многим. Выведите список всех оркестров, у которых название начинается с буквы «О», и имена их дирижёров.

Текст программы:

class Conductor:

    # Дирижер

    def \_\_init\_\_(self, id, name, age, salary):

        self.id = id

        self.name = name

        self.age = age

        self.salary = salary

    def \_\_str\_\_(self):

        return f'ID: {self.id}, Имя: {self.name}, Возраст: {self.age}, Зарплата: {self.salary}'

    def \_\_repr\_\_(self):

        return self.\_\_str\_\_()

class Orchestra:

    # Оркестр

    def \_\_init\_\_(self, id, name, conductorid, creationdate):

        self.id = id

        self.name = name

        self.conductorid = conductorid

        self.creationdate = creationdate

    def \_\_str\_\_(self):

        return f'ID: {self.id}, Имя: {self.name}, ID дирижера: {self.conductorid}, Дата создания: {self.creationdate}'

    def \_\_repr\_\_(self):

        return self.\_\_str\_\_()

class CondOrch:

    # Связующая таблица

    def \_\_init\_\_(self, conductorid, orchestraid):

        self.conductorid = conductorid

        self.orchestraid = orchestraid

# Дирижеры . Отдел

conds = [

    Conductor(1, 'Иванов', 45, 100000),

    Conductor(2, 'Петров', 50, 120000),

    Conductor(3, 'Сидоров', 55, 150000),

    Conductor(4, 'Смирнов', 60, 200000),

    Conductor(5, 'Александров', 65, 250000),

    Conductor(6, 'Попов', 70, 300000),

    Conductor(7, 'Козлов', 75, 350000),

    Conductor(8, 'Лебедев', 80, 400000),

    Conductor(9, 'Новиков', 85, 450000),

    Conductor(10, 'Алексеев', 90, 500000),

    Conductor(111, 'Морозов', 45, 100000),

    Conductor(222, 'Волков', 50, 120000),

    Conductor(333, 'Кузнецов', 55, 150000),

]

# Оркестры . Сотрудники

orches = [

    Orchestra(1, 'Оркестр им. Шостаковича', 1, 1990),

    Orchestra(2, 'Оркестр им. Чайковского', 2, 1995),

    Orchestra(3, 'Оркестр им. Бетховена', 3, 2000),

    Orchestra(4, 'Ансамбль им. Моцарта', 4, 2005),

    Orchestra(5, 'Марьячи им. Баха', 5, 2010),

    Orchestra(6, 'Тараф им. Вивальди', 6, 2015),

    Orchestra(7, 'Оркестр им. Барби', 7, 2020),

    Orchestra(8, 'Квартет им. Брамса', 8, 2025),

    Orchestra(9, 'Октет им. Бритен', 9, 2030),

    Orchestra(10, 'Оркестр им. Берлиоза', 10, 2035),

    Orchestra(11, 'Оркестр им. Бизе', 10, 2040),

    Orchestra(12, 'Оркестр им. Бородина', 10, 2045),

    Orchestra(13, 'Оркестр Московского театра оперы и балета им. А.С. Пушкина', 10, 2050),

]

# Связующая таблица

condorch = [

    CondOrch(1, 1),

    CondOrch(2, 2),

    CondOrch(3, 3),

    CondOrch(4, 4),

    CondOrch(5, 5),

    CondOrch(6, 6),

    CondOrch(7, 7),

    CondOrch(8, 8),

    CondOrch(9, 9),

    CondOrch(10, 10),

    CondOrch(10, 11),

    CondOrch(10, 12),

    CondOrch(10, 13),

    CondOrch(111, 1),

    CondOrch(222, 2),

    CondOrch(333, 3),

    CondOrch(333, 4),

    CondOrch(333, 5),

]

def main():

    one\_to\_many = [

        (c.name, c.age, o.name, o.creationdate)

        for c in conds

        for o in orches

        if c.id == o.conductorid

    ]

    print("Задание Е1")

    ans = [

        {

            c.name: [

                (o[2], o[3]) for o in one\_to\_many if c.name == o[0]

            ]

        } for c in conds if 'алекс' in c.name.lower()

    ]

    print(ans)

    print()

    print("Задание Е2")

    ans = sorted([

        (c.name, round(sum([o[3] for o in one\_to\_many if c.name == o[0]]) /

         len([o[3] for o in one\_to\_many if c.name == o[0]]), 2))

        for c in conds if len([o[3] for o in one\_to\_many if c.name == o[0]]) != 0

    ], key=lambda x: x[1], reverse=True)

    print(ans)

    print()

    "«Дирижер» и «Оркестр» связаны соотношением многие-ко-многим. Выведите список всех оркестров, у которых название начинается с буквы «О», и имена их дирижёров."

    print("Задание Е3")

    many\_to\_many = [

        (c.name, c.age, c.salary, o.name, o.creationdate)

        for c in conds

        for o in orches

        for co in condorch

        if c.id == co.conductorid and o.id == co.orchestraid

    ]

    ans = {

        o[3]: [

            (c[0], c[1], c[2]) for c in many\_to\_many if o[3] == c[3]

        ] for o in many\_to\_many if o[3][0] == 'О'

    }

    print(ans)

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

    main()

Результаты выполнения:

PS D:\projects\bmstu\_learning> python -u "d:\projects\bmstu\_learning\rk1\krainikov.rk1.py"

Задание Е1

[{'Александров': [('Марьячи им. Баха', 2010)]}, {'Алексеев': [('Оркестр им. Берлиоза', 2035), ('Оркестр им. Бизе', 2040), ('Оркестр им. Бородина', 2045), ('Оркестр Московского театра оперы и балета им. А.С. Пушкина', 2050)]}]

Задание Е2

[('Алексеев', 2042.5), ('Новиков', 2030.0), ('Лебедев', 2025.0), ('Козлов', 2020.0), ('Попов', 2015.0), ('Александров', 2010.0), ('Смирнов', 2005.0), ('Сидоров', 2000.0), ('Петров', 1995.0), ('Иванов', 1990.0)]

Задание Е3

{'Оркестр им. Шостаковича': [('Иванов', 45, 100000), ('Морозов', 45, 100000)], 'Оркестр им. Чайковского': [('Петров', 50, 120000), ('Волков', 50, 120000)], 'Оркестр им. Бетховена': [('Сидоров', 55, 150000), ('Кузнецов', 55, 150000)], 'Оркестр им. Барби': [('Козлов', 75, 350000)], 'Октет им. Бритен': [('Новиков', 85, 450000)], 'Оркестр им. Берлиоза': [('Алексеев', 90, 500000)], 'Оркестр им. Бизе': [('Алексеев', 90, 500000)], 'Оркестр им. Бородина': [('Алексеев', 90, 500000)], 'Оркестр Московского театра оперы и балета им. А.С. Пушкина': [('Алексеев', 90, 500000)]}