

**Московский государственный технический
университет им. Н.Э. Баумана**

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

Курс «Парадигмы и конструкции языков программирования»

РК №1

Вариант №5(Г)

Выполнил:
студент группы ИУ5-24Б:
Здобняков Фёдор
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Правдина Ю.Е.
Подпись и дата:

Москва, 2023 г.

Условия рубежного контроля №1 по курсу Пик ЯП

Задание:

Рубежный контроль представляет собой разработку программы на языке Python, которая выполняет следующие действия:

1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.

Пример классов данных для предметной области Сотрудник-Отдел:

1. Класс «Сотрудник», содержащий поля:
 - ID записи о сотруднике;
 - Фамилия сотрудника;
 - Зарплата (количественный признак);
 - ID записи об отделе. (для реализации связи один-ко-многим)
 2. Класс «Отдел», содержащий поля:
 - ID записи об отделе;
 - Наименование отдела.
 3. (Для реализации связи многие-ко-многим) Класс «Сотрудники отдела», содержащий поля:
 - ID записи о сотруднике;
 - ID записи об отделе.
- 2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
- 3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Для реализации запроса №2 введите в класс, находящийся на стороне связи «много», произвольный количественный признак, например, «зарплата сотрудника».

Результатом рубежного контроля является документ в формате PDF, который содержит текст программы и результаты ее выполнения.

Вариант Г.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

5	Музыкант	Оркестр
---	----------	---------

Код программы:

```
mus_ID = 1
orc_ID = 1

class Orchestra:
    def __init__(self, name):
        global orc_ID
        self.name = name
        self.orc_id = orc_ID
        orc_ID += 1

class Musician:
    def __init__(self, name, salary, orc_ID):
        global mus_ID
        self.name = name
        self.salary = salary
        self.mus_ID = mus_ID
        self.orc_ID = orc_ID
        mus_ID += 1

class Mus_In_Orc:
    def __init__(self, mus_ID, orc_ID):
        self.mus_ID = mus_ID
        self.orc_ID = orc_ID

mus = []
orc = []
m_o = []

with open("musician.txt", "r") as file:
    for line in file:
        data = line.split()
        mus1 = Musician(data[0], int(data[1]), int(data[2]))
        mus.append(mus1)

with open("Orchestra.txt", "r") as file:
    for line in file:
```

```

        data = line.split()
        orc1 = Orchestra(data[0])
        orc.append(orc1)

for i in orc:
    for j in mus:
        if i.orc_id == j.orc_ID:
            m_o1 = Mus_In_Orc(j.mus_ID, i.orc_id)
            m_o.append(m_o1)

#Задание 1
print("Задание 1")
for i in orc:
    if i.name[0] == 'A':
        print(i.name)
        print("Сотрудники:")
        for j in mus:
            if i.orc_id == j.orc_ID:
                print(j.name)

print('\n', '\n')
print("Задание 2")
#Задание 2
sp = []
for i in orc:
    max_sal = 0
    for j in mus:
        if j.orc_ID == i.orc_id and j.salary > max_sal:
            max_sal = j.salary

    sp.append([i.name, max_sal])

sp.sort(key=lambda arr: arr[1])
sp = sp
for i in sp:
    print(i[0], "максимальная зарплата:", i[1])

print("Отсортировано по возрастанию")

#Задание 3
print('\n', '\n')
print("Задание 3")

for i in orc:
    print("Сотрудники, работающие в оркестре", i.name, ':')
    for j in mus:
        if j.orc_ID == i.orc_id:
            print(j.name)
    print('')

```

Текстовые базы данных:

1	Aria_Orchestra
2	Estrad_Orchestra
3	Jazz_Orchestra
4	
1	Василий 30000 2
2	Сергей 40000 1
3	Виктор 20000 3
4	Никита 50000 2
5	Дмитрий 60000 1

Результаты:

```
/Users/Fedor/PycharmProjects/2kurs/venv/bin/python /Users/Fedor/PycharmProjects/2kurs/RK.py
Задание 1
Aria_Orchestra
Сотрудники:
Сергей
Дмитрий

Задание 2
Jazz_Orchestra максимальная зарплата: 20000
Estrad_Orchestra максимальная зарплата: 50000
Aria_Orchestra максимальная зарплата: 60000
Отсортировано по возрастанию

Задание 3
Сотрудники, работающие в оркестре Aria_Orchestra :
Сергей
Дмитрий

Сотрудники, работающие в оркестре Estrad_Orchestra :
Василий
Никита

Сотрудники, работающие в оркестре Jazz_Orchestra :
Виктор

Process finished with exit code 0
|
```