

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

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

Курс «Разработка интернет приложений»

Рубежный контроль №1

Вариант №2Б

Выполнил:

студент группы ИУ5-52Б
Бабин Артём

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

Описание задания:

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

Вариант Б.

1. «Класс» и «Школьник» связаны соотношением один-ко-многим. Выведите список всех связанных школьников и классов, отсортированный по школьникам, сортировка по классам произвольная.
2. «Класс» и «Школьник» связаны соотношением один-ко-многим. Выведите список классов с количеством школьников в каждом классе, отсортированный по количеству школьников.
3. «Класс» и «Учитель» связаны соотношением многие-ко-многим. Выведите список всех учителей, у которых фамилия заканчивается на «вич», и названия их классов.

Текст программы

```
class Class:
    def __init__(self, id, number, letter):
        self.id = id
        self.number = number
        self.letter = letter

    def full_name(self):
        return f'{self.number}{self.letter}'

class Student:
    def __init__(self, id, first_name, last_name, age, id_class):
```

```

        self.id = id
        self.first_name = first_name
        self.last_name = last_name
        self.age = age
        self.id_class = id_class

    def full_name(self):
        return f'{self.first_name} {self.last_name}'

class Teacher:
    def __init__(self, id, fio):
        self.id = id
        self.fio = fio

class ClassTeacher:
    def __init__(self, id, id_class, id_teacher):
        self.id = id
        self.id_class = id_class
        self.id_teacher = id_teacher

def main():
    classes = [Class(0, 1, 'А'),
               Class(1, 1, 'Б'),
               Class(2, 2, 'А')]

    students = [Student(0, 'Иван', 'Иванов', 8, 1),
                Student(1, 'Пётр', 'Петров', 9, 2),
                Student(2, 'Артём', 'Бабин', 9, 1),
                Student(3, 'Анна', 'Перова', 10, 2),
                Student(4, 'Михаил', 'Барышников', 8, 2),
                Student(4, 'Дарья', 'Васильченко', 8, 0)]

    teachers = [Teacher(0, 'Козлов Александр Дмитриевич'),
                Teacher(1, 'Ахметова Фания Харисовна'),
                Teacher(2, 'Скрипниченко Пётр Петрович')]

    classes_teachers = [ClassTeacher(0, 0, 0),
                        ClassTeacher(0, 0, 1),
                        ClassTeacher(0, 1, 0),
                        ClassTeacher(0, 1, 1),

```

```

        ClassTeacher(0, 1, 2),
        ClassTeacher(0, 2, 2)]

    # part1
    res1 = sorted([(stud.full_name(), cls.full_name()) for stud in
students for cls in classes if stud.id_class == cls.id], key=lambda x:
x[0])
    res2 = sorted({cls.full_name(): len(list(filter(lambda x: x.id_class
== cls.id, students))) for cls in classes}.items(), key=lambda x: x[0],
reverse=True)
    res3 = {teach.fio: [cls.full_name() for cls in classes if cls.id in
[cls_teach.id_class for cls_teach in classes_teachers if
cls_teach.id_teacher == teach.id]] for teach in teachers if
str(teach.fio).endswith('вич')}]
    print(res1)
    print(res2)
    print(res3)

if __name__ == '__main__':
    main()

```

Экранная форма с результатом выполнения программы:

```

[('Анна Перова', '2А'), ('Артём Бабин', '1Б'), ('Дарья Васильченко', '1А'), ('Иван Иванов', '1Б'), ('Михаил Барышников', '2А'), ('Пётр Петров', '2А')]
[('2А', 3), ('1Б', 2), ('1А', 1)]
{'Козлов Александр Дмитриевич': ['1А', '1Б'], 'Скрипниченко Пётр Петрович': ['1Б', '2А']}

```