Московский государственный технический университет им. Н.Э. Баумана

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

> Курс «Разработка интернет приложений» Отчет по рубежному контролю №1

Выполнил:

студент группы ИУ5-53Б

Алёшин Александр

Вариант РК:

Вариант В.

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

Вариант запросов РК1	Вариант предметной области РК	
В	2	

№ варианта	Класс 1	Класс 2
2	Школьник	Класс

Код программы:

```
class Pup:
    """YueHUK"""

    def __init__(self, id, fio, age, cls_id):
        self.id = id
        self.cls_id = cls_id
        self.age = age

class Cls:
    """Kfacc"""

    def __init__(self, id, name):
        self.id = id
        self.name = name
```

```
'Ученики класса' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    def init (self, cls id, pup id):
        self.cls_id = cls_id
        self.pup id = pup id
# классы
classes = [
    Cls(1, '11 A'),
    Cls(2, '11 B'),
    Cls(3, '10 A'),
    Cls(11, '1 A'),
    Cls(22, '1 B'),
    Cls(33, '1 B'),
]
# ученики
pupils = [
    Рир (1, 'Артамонов', 18, 1),
    Рир (2, 'Петров', 17, 1),
    Рир (3, 'Иваненко', 16, 2),
    Рир (4, 'Иванов', 17, 2),
    Рир (5, 'Иванин', 16, 3),
    Pup (6, "Фёдоров", 15, 3)
1
cls pups = [
    ClsPup(1, 1),
    ClsPup(1, 2),
    ClsPup(2, 3),
    ClsPup(2, 4),
    ClsPup(3, 5),
    ClsPup(3, 6),
]
def main():
    """Основная функция"""
    # Соединение данных один-ко-многим
    one to many = [(e.fio, e.age, d.name)
                   for d in classes
                   for e in pupils
                   if e.cls id == d.id]
    # Соединение данных многие-ко-многим
    many to many temp = [(d.name, ed.cls id, ed.pup id)
                         for d in classes
                         for ed in cls pups
                         if d.id == ed.cls id]
    many to many = [(e.fio, cls name)
                    for cls_name, cls_id, pup_id in many_to_many_temp
                    for e in pupils if e.id == pup_id]
```

```
sort = list(filter(lambda x: x[0].startswith('A'), one to many))
   print('Задание B1')
   res 11 = sorted(sort, key=itemgetter(1))
   print(res 11)
   print('\nЗадание B2')
   res 12 unsorted = []
    # Перебираем все отделы
    for d in classes:
        # Список учеников класса
        d pups = list(filter(lambda i: i[2] == d.name, one to many))
       # Если класс не пустой
       if len(d pups) > 0:
            # Возраст учеников класса
            d_age = [age for _, age, _ in d_pups]
            # Минимальный возраст учеников класса
            d_age_min = min(d age)
            res 12 unsorted.append((d.name, d age min))
    # Сортировка по минимальному возрасту
   res 12 = sorted(res 12 unsorted, key=itemgetter(1))
   print(res 12)
   print('\nЗадание ВЗ')
   res 13 = {}
    # Перебираем все
   print(sorted(many to many, key=itemgetter(0)))
if __name__ == '__main__':
   main()
```

Результат работы программы:

```
Задание B1
[('Артамонов', 18, '11 A')]

Задание B2
[('10 A', 15), ('11 5', 16), ('11 A', 17)]

Задание B3
[('Артамонов', '11 A'), ('Иваненко', '11 5'), ('Иванин', '10 A'), ('Иванов', '11 Б'), ('Петров', '11 A'), ('Фёдоров', '10 A')]
{}

Process finished with exit code 0
```