

```

from collections import Counter
from collections import defaultdict

class Teacher:
    """Преподаватель"""
    def __init__(self, teacher_id, fio, sal, course_id):
        self.teacher_id = teacher_id
        self.fio = fio
        self.sal = sal
        self.course_id = course_id

class Course:
    """Учебный курс"""
    def __init__(self, course_id, name):
        self.course_id = course_id
        self.name = name

class TeacherCourse:
    """
    'Преподаватели учебного курса' для реализации
    связи многие-ко-многим
    """
    def __init__(self, teacher_id, course_id):
        self.teacher_id = teacher_id
        self.course_id = course_id

# Учебные курсы
Courses = [
    Course(1, 'Теория вероятностей'),
    Course(2, 'Модели данных'),
    Course(3, 'Правоведение')
]

# Преподаватели
Teachers = [
    Teacher(1, 'Куликов', 75000, 1),
    Teacher(2, 'Мальцев', 105000, 1),
    Teacher(3, 'Борисенко', 93000, 2),
    Teacher(4, 'Кузнецов', 78000, 3),
    Teacher(5, 'Аксенов', 81000, 3),
]

Teachers_Courses = [
    TeacherCourse(1,1),
    TeacherCourse(2,1),
    TeacherCourse(3,2),
    TeacherCourse(4,3),
    TeacherCourse(5,3)
]

def main():

```

```
print('\nЗапрос 1: список всех преподавателей и их учебных курсов, отсортированных по фамилии преподавателя\n')
```

```
sorted_Teachers = sorted(Teachers, key=lambda x: x.fio)
```

```
course_dict = {carpark.course_id: carpark.name for carpark in Courses}
```

```
[print(f"Преподаватель: {Teacher.fio}, Учебный курс: {course_dict.get(Teacher.course_id, 'Неизвестный учебный кур')}") for Teacher in sorted_Teachers]
```

```
print('\nЗапрос 2: список учебных курсов и количества преподавателей в них, отсортированный по количеству преподавателей\n')
```

```
num_Teachers = Counter(d.course_id for d in Teachers)
```

```
sorted_carparks = sorted(Courses, key=lambda x: num_Teachers[x.course_id], reverse=True)
```

```
[print(f"Учебный курс: {carpark.name}, Количество преподавателей: {num_Teachers[carpark.course_id]}") for carpark in sorted_carparks]
```

```
print('\nЗапрос 3: список преподавателей, фамилия которых заканчивается на "ов" и их учебных курсов\n')
```

```
Teacher_course_dict = defaultdict(list)
```

```
for cd in Teachers_Courses:
    Teacher_course_dict[cd.teacher_id].append(cd.course_id)
```

```
filtered_Teachers = [Teacher for Teacher in Teachers if Teacher.fio.endswith("ов")]
```

```
[print(f"Преподаватель: {Teacher.fio}, Учебный(е) курс(ы): {'.'.join(course_dict[course_id] for course_id in Teacher_course_dict.get(Teacher.teacher_id, []))}") for Teacher in filtered_Teachers]
```

```
if __name__ == '__main__':
    main()
```

Результат выполнения:

Запрос 1: список всех преподавателей и их учебных курсов, отсортированных по фамилии преподавателя

Преподаватель: Аксенов, Учебный курс: Правоведение

Преподаватель: Борисенко, Учебный курс: Модели данных

Преподаватель: Кузнецов, Учебный курс: Правоведение

Преподаватель: Куликов, Учебный курс: Теория вероятностей

Преподаватель: Мальцев, Учебный курс: Теория вероятностей

Запрос 2: список учебных курсов и количества преподавателей в них, отсортированный по количеству преподавателей

Учебный курс: Теория вероятностей, Количество преподавателей: 2

Учебный курс: Правоведение, Количество преподавателей: 2

Учебный курс: Модели данных, Количество преподавателей: 1

Запрос 3: список преподавателей, фамилия которых заканчивается на "ов" и их учебных курсов

Преподаватель: Куликов, Учебный(е) курс(ы): Теория вероятностей

Преподаватель: Кузнецов, Учебный(е) курс(ы): Правоведение

Преподаватель: Аксенов, Учебный(е) курс(ы): Правоведение