

## РК 1

### Вариант Д.

- 1) «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.
- 2) «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате (отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений).
- 3) «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.

### Адаптация задания под Вариант 1 (Студент, Группа)

- 1) «Группа» и «Студент» связаны соотношением один-ко-многим. Выведите список всех Студентов, у которых фамилия заканчивается на «ов», и названия их Групп.
- 2) «Группа» и «Студент» связаны соотношением один-ко-многим. Выведите список Групп со средним количеством баллов Студентов в каждой группе, отсортированный по среднему количеству баллов.
- 3) «Группа» и «Студент» связаны соотношением многие-ко-многим. Выведите список всех Групп, у которых название начинается с буквы «А», и список учащихся в них студентов.

### Выполнение:

```
class Student:
```

```
    def __init__(self, student_id, last_name, points, group_id):
        self.student_id = student_id
        self.last_name = last_name
        self.points = points
        self.group_id = group_id
```

```
class Group:
```

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

```
class StudentsGroups:
```

```
    def __init__(self, student_id, group_id):
        self.student_id = student_id
        self.group_id = group_id
```

```

groups = [
    Group(1, "А Группа"),
    Group(2, "В Группа"),
    Group(3, "С Группа"),
    Group(4, "D Группа"),
    Group(5, "A+ Группа")
]
students = [
    Student(1, "Иванов", 85, 1),
    Student(2, "Петров", 90, 1),
    Student(3, "Сидоров", 75, 2),
    Student(4, "Алексеев", 88, 3),
    Student(5, "Андреев", 92, 3),
    Student(7, "Андреев", 100, 5),
    Student(6, "Федоров", 80, 4)
]
students_groups = [
    StudentsGroups(1, 1),
    StudentsGroups(2, 1),
    StudentsGroups(3, 2),
    StudentsGroups(4, 3),
    StudentsGroups(5, 3),
    StudentsGroups(6, 4),
    StudentsGroups(7, 5)
]

def get_students_with_lastname_ending_with_ov():
    result = []
    for student in students:
        if student.last_name.endswith("ов"):
            group_name = next((group.name for group in groups if group.group_id ==
student.group_id), None)
            result.append((student.last_name, group_name))
    return result

students_with_ov = get_students_with_lastname_ending_with_ov()
print(students_with_ov, '\n')

def get_average_points_by_group():
    group_points = {}
    for student in students:
        if student.group_id not in group_points:
            group_points[student.group_id] = []
        group_points[student.group_id].append(student.points)

```

```

average_points = []

for group in groups:
    points = group_points.get(group.group_id, [])
    if points:
        total_points = sum(points)
        count_points = len(points)
        avg_points = total_points / count_points
        average_points.append((group.name, avg_points))

return sorted(average_points, key=lambda x: x[1])

average_points = get_average_points_by_group()
print(average_points, '\n')

def get_students_in_groups_starting_with_A():
    result = {}

    for group in groups:
        if group.name.startswith("A"):
            enrolled_students = [
                student.last_name for sg in students_groups
                for student in students
                if sg.group_id == group.group_id and sg.student_id ==
student.student_id
            ]
            result[group.name] = enrolled_students

    return result

students_in_groups_A = get_students_in_groups_starting_with_A()
print(students_in_groups_A, '\n')

```

### **Результаты:**

№1

```
[('Иванов', 'А Группа'), ('Петров', 'А Группа'), ('Сидоров', 'В Группа'),
('Федоров', 'D Группа')]
```

№2

```
[('В Группа', 75.0), ('D Группа', 80.0), ('А Группа', 87.5), ('С Группа', 90.0), ('А+
Группа', 100.0)]
```

№3

```
{'А Группа': ['Иванов', 'Петров'], 'А+ Группа': ['Андреев']}
```