

## main.py

```
from Department import Department
from StudentGroup import StudentGroup
from StudentGroupDepartment import StudentGroupDepartment
from Requests import findGroupsEndWithKey,
DepartmentsSortedByAvgNumberOfStudents, findDepartmentsStartWithKey

# Список кафедр
departments = [
    Department(id=1, name="Прикладная математика"),
    Department(id=2, name="Информатика"),
    Department(id=3, name="Физика"),
    Department(id=4, name="Астрономия"),
    Department(id=5, name="История"),
    Department(id=6, name="Археология"),
    Department(id=7, name="Астрофизика"),
]

# Список студенческих групп
student_groups = [
    StudentGroup(id=1, group_name="Группа 101", number_of_students=30,
department_id=1),
    StudentGroup(id=2, group_name="Группа 102", number_of_students=25,
department_id=2),
    StudentGroup(id=3, group_name="Группа 103", number_of_students=28,
department_id=1),
    StudentGroup(id=4, group_name="Группа 104", number_of_students=22,
department_id=3),
    StudentGroup(id=5, group_name="Группа 105", number_of_students=27,
department_id=4),
    StudentGroup(id=6, group_name="Группа 1003", number_of_students=20,
department_id=4),
    StudentGroup(id=7, group_name="Группа 104А", number_of_students=21,
department_id=5),
    StudentGroup(id=8, group_name="Группа 105А", number_of_students=24,
department_id=6),
    StudentGroup(id=9, group_name="Группа 1003А", number_of_students=23,
department_id=7),
    StudentGroup(id=10, group_name="Группа 104Б", number_of_students=21,
department_id=5),
    StudentGroup(id=11, group_name="Группа 145Б", number_of_students=20,
department_id=4),
    StudentGroup(id=12, group_name="Группа 1Б03", number_of_students=19,
department_id=7),
]

group_deps = [
    StudentGroupDepartment(student_group_id=1, department_id=1),
    StudentGroupDepartment(student_group_id=2, department_id=2),
    StudentGroupDepartment(student_group_id=3, department_id=1),
    StudentGroupDepartment(student_group_id=4, department_id=3),
    StudentGroupDepartment(student_group_id=5, department_id=4),
    StudentGroupDepartment(student_group_id=6, department_id=4),
    StudentGroupDepartment(student_group_id=7, department_id=5),
    StudentGroupDepartment(student_group_id=8, department_id=6),
    StudentGroupDepartment(student_group_id=9, department_id=7),
    StudentGroupDepartment(student_group_id=10, department_id=5),
    StudentGroupDepartment(student_group_id=11, department_id=4),
    StudentGroupDepartment(student_group_id=12, department_id=7),
]

# Таблица для отношения один-ко-многим
```

```

one_to_many = [(g.group_name, g.number_of_students, d.name)
                for d in departments
                for g in student_groups
                if g.department_id == d.id]

# Соединение таблицы многие-ко-многим
temp = [(group.group_name, group_dep.department_id)
        for group in student_groups
        for group_dep in group_deps
        if group.id == group_dep.student_group_id]

many_to_many = [(group_name, dept.name)
                 for dept in departments
                 for group_name, dep_id in temp
                 if dep_id == dept.id]

if __name__ == '__main__':
    # Запрос 1: Студенческие группы, заканчивающиеся на key и их кафедры
    (один-ко-многим)
    findGroupsEndWithKey(one_to_many, "03")

    # Запрос 2: Кафедры, отсортированные по среднему кол-ву студентов в
    группах (один-ко-многим)
    DepartmentsSortedByAvgNumberOfStudents(one_to_many)

    # Запрос 3: Кафедры, начинающиеся на букву == key и их студенческие
    группы (многие-ко-многим)
    findDepartmentsStartWithKey(many_to_many, "A")

```

### StudentGroupDepartment.py

```

class StudentGroupDepartment:
    def __init__(self, student_group_id, department_id):
        self.student_group_id = student_group_id
        self.department_id = department_id

    def __repr__(self):
        return
        f"StudentGroupDepartment(student_group_id={self.student_group_id},
        department_id={self.department_id})"

```

### StudentGroup.py

```

class StudentGroup:
    def __init__(self, id, group_name, number_of_students, department_id):
        self.id = id
        self.group_name = group_name
        self.number_of_students = number_of_students
        self.department_id = department_id

    def __repr__(self):
        return (f"StudentGroup(id={self.id}, group_name='{self.group_name}',
        "
                f"number_of_students={self.number_of_students},
        department_id={self.department_id})")

```

## Department.py

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

    def __repr__(self):
        return f"Department(id={self.id}, name='{self.name}')
```

## Requests.py

```
from collections import defaultdict

"""Запрос 1: Студенческие группы, заканчивающиеся на key и их кафедры"""
def findGroupsEndWithKey(one_to_many, key, debug=True):
    # Фильтруем группы, названия которых заканчиваются на ключ
    filtered_groups = [
        {"group_name": group_name, "department_name": department_name}
        for group_name, _, department_name in one_to_many
        if group_name.endswith(key)
    ]

    if debug:
        print(f"Запрос 1: Студенческие группы, заканчивающиеся на '{key}' и их кафедры:")
        if not filtered_groups:
            print("Ничего не найдено")
        else:
            for item in filtered_groups:
                print(f"Группа: {item['group_name']}, Кафедра: {item['department_name']}")
            print("\n")

    return filtered_groups

"""Запрос 2: Кафедры, отсортированные по среднему кол-ву студентов в группах"""
def DepartmentsSortedByAvgNumberOfStudents(one_to_many, debug=True):
    # Сгруппируем количество студентов по названиям кафедр
    department_student_counts = defaultdict(list)
    for group_name, number_of_students, department_name in one_to_many:
        department_student_counts[department_name].append(number_of_students)

    # Вычислим среднее количество студентов для каждой кафедры
    average_students = [
        {
            "department_name": dept,
            "average_number_of_students": sum(counts) / len(counts)
        }
        for dept, counts in department_student_counts.items()
    ]

    # Отсортируем кафедры по среднему количеству студентов в порядке убывания
    sorted_average_students = sorted(
        average_students,
        key=lambda x: x["average_number_of_students"],
        reverse=True
    )

    if debug:
```

```

        print("Запрос 2: Кафедры со средним количеством студентов,
отсортированные по среднему количеству:")
        for item in sorted_average_students:
            print(
                f"Кафедра: {item['department_name']}, "
                f"Среднее количество студентов: "
                f"{item['average_number_of_students']:.2f}"
            )
        print("\n")

    return sorted_average_students

"""Запрос 3: Кафедры, начинающиеся на букву == key и их студенческие
группы"""
def findDepartmentsStartWithKey(many_to_many, key, debug=True):
    # Сгруппируем группы по названиям кафедр
    department_to_groups = defaultdict(list)
    for group_name, department_name in many_to_many:
        department_to_groups[department_name].append(group_name)

    # Фильтруем кафедры, названия которых начинаются с ключа
    filtered_departments = {
        dept: groups
        for dept, groups in department_to_groups.items()
        if dept.startswith(key)
    }

    result = [
        {"department_name": dept, "student_groups": groups}
        for dept, groups in filtered_departments.items()
    ]

    if debug:
        print(f"Запрос 3: Кафедры, начинающиеся на '{key}', и их студенческие
группы:")
        if not result:
            print("Ничего не найдено")
        else:
            for item in result:
                groups = "\n\t".join(item["student_groups"]) if
item["student_groups"] else "Нет групп"
                print(f"Кафедра {item['department_name']}:\n\t{groups}")
            print("\n")

    return result

```

## Результаты выполнения:

Запрос 1: Студенческие группы, заканчивающиеся на '03' и их кафедры:

Группа: Группа 103, Кафедра: Прикладная математика

Группа: Группа 1003, Кафедра: Астрономия

Группа: Группа 1503, Кафедра: Астрофизика

Запрос 2: Кафедры со средним количеством студентов, отсортированные по среднему количеству:

Кафедра: Прикладная математика, Среднее количество студентов: 29.00

Кафедра: Информатика, Среднее количество студентов: 25.00

Кафедра: Археология, Среднее количество студентов: 24.00

Кафедра: Астрономия, Среднее количество студентов: 22.33

Кафедра: Физика, Среднее количество студентов: 22.00

Кафедра: История, Среднее количество студентов: 21.00

Кафедра: Астрофизика, Среднее количество студентов: 21.00

Запрос 3: Кафедры, начинающиеся на 'А', и их студенческие группы:

Кафедра Астрономия:

Группа 105

Группа 1003

Группа 145Б

Кафедра Археология:

Группа 105А

Кафедра Астрофизика:

Группа 1003А

Группа 1503

Запрос 1: Студенческие группы, заканчивающиеся на 'Б' и их кафедры:

Группа: Группа 145Б, Кафедра: Астрономия

Группа: Группа 104Б, Кафедра: История

Запрос 2: Кафедры со средним количеством студентов, отсортированные по среднему количеству:

Кафедра: Прикладная математика, Среднее количество студентов: 29.00

Кафедра: Информатика, Среднее количество студентов: 25.00

Кафедра: Археология, Среднее количество студентов: 24.00

Кафедра: Астрономия, Среднее количество студентов: 22.33

Кафедра: Физика, Среднее количество студентов: 22.00

Кафедра: История, Среднее количество студентов: 21.00

Кафедра: Астрофизика, Среднее количество студентов: 21.00

Запрос 3: Кафедры, начинающиеся на 'И', и их студенческие группы:

Кафедра Информатика:

Группа 102

Кафедра История:

Группа 104А

Группа 104Б

Запрос 1: Студенческие группы, заканчивающиеся на '9' и их кафедры:  
Ничего не найдено

Запрос 2: Кафедры со средним количеством студентов, отсортированные по среднему количеству:  
Кафедра: Прикладная математика, Среднее количество студентов: 29.00  
Кафедра: Информатика, Среднее количество студентов: 25.00  
Кафедра: Археология, Среднее количество студентов: 24.00  
Кафедра: Астрономия, Среднее количество студентов: 22.33  
Кафедра: Физика, Среднее количество студентов: 22.00  
Кафедра: История, Среднее количество студентов: 21.00  
Кафедра: Астрофизика, Среднее количество студентов: 21.00

Запрос 3: Кафедры, начинающиеся на 'Л', и их студенческие группы:  
Ничего не найдено