



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

Факультет «ГУИМЦ»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»»

ОТЧЕТ

ПО РУБЕЖНОМУ КОНТРОЛЮ №1

Вариант предметной области 28

Вариант запросов: В

Студент: Агапова А.Д., группа ИУ5Ц-52Б

Преподаватель: Гапанюк Ю. Е.

2024г.

Вариант запросов В. Предметная область 28

1 «Кафедра» и «Студенческая группа» связаны соотношением один-ко-многим.

Выведите список всех групп, относящихся к факультету ИУ (название группы начинается с «ИУ»), и названия их кафедр.

2 «Кафедра» и «Студенческая группа» связаны соотношением один-ко-многим.

Выведите список кафедр с максимальным количеством людей в группе на каждой кафедре, отсортированный по минимальному количеству людей.

3 «Кафедра» и «Студенческая группа» связаны соотношением многие-ко-многим. Выведите список всех связанных групп и кафедр, отсортированный по группам, сортировка по кафедрам произвольная.

Листинг программы

```
class Group:
    def __init__(self, id: int, department: str, sem: int, index: int,
faculty_id: int, students: int):
        self._id = id
        self._name = f"{department}-{sem}{index}"
        self._faculty_id = faculty_id
        self._students = students

    @property
    def id(self) -> int:
        return self._id

    @property
    def faculty_id(self) -> int:
        return self._faculty_id

    @property
    def name(self) -> str:
        return self._name

    @property
    def students(self) -> int:
        return self._students

class Faculty:
    def __init__(self, id: int, title: str):
        self._id = id
        self._title = title

    @property
    def id(self) -> int:
        return self._id

    @property
    def title(self) -> str:
        return self._title

class GroupFaculty:
    def __init__(self, group_id: int, faculty_id: int):
        self._group_id = group_id
```

```

        self._faculty_id = faculty_id

    @property
    def faculty_id(self) -> int:
        return self._faculty_id

    @property
    def group_id(self) -> int:
        return self._group_id

def task1(faculties: list[Faculty], groups: list[Group]):
    print("Запрос № 1")
    data = [(g, c)
              for g in groups
              for c in faculties
              if (g.faculty_id == c.id and g.name.startswith("ИУ")) ]
    for (g, c) in data:
        print(g.name, c.title)
    print()

def task2(faculties: list[Faculty], groups: list[Group]):
    print("Запрос № 2")
    data = {}
    for faculty in faculties:
        faculty_students = [g.students
                             for g in groups
                             for c in faculties
                             if (g.faculty_id == c.id and c.id == faculty.id)
        ]
        data[faculty.title] = max(faculty_students)

    data_items = list(data.items())
    data_items.sort(key = lambda x: x[1])
    for (faculty, max_students) in data_items:
        print(faculty, max_students)
    print()

def task3(faculties: list[Faculty], groups: list[Group], groups_faculties:
list[GroupFaculty]):
    print("Запрос № 3")
    data = [(g, f)
              for gf in groups_faculties
              for g in groups
              for f in faculties
              if gf.group_id == g.id and gf.faculty_id == f.id]

    data.sort(key = lambda x: x[0].name)

    for (group, course) in data:
        print(group.name, course.title)
    print()

def main():
    faculties = [
        Faculty(1, "Информатика и системы управления"),
        Faculty(2, "Инженерный бизнес и менеджмент"),
        Faculty(3, "Робототехника и комплексная автоматизация"),
        Faculty(4, "Фундаментальные науки"),
        Faculty(5, "Энергомашиностроение")
    ]

    groups = [

```

```

        Group(1, "ИУ5ц", 5, 1, 1, 28),
        Group(2, "ИУ6ц", 5, 2, 1, 24),
        Group(3, "ИУ7", 3, 3, 1, 29),
        Group(4, "ИБМ7", 3, 4, 2, 27),
        Group(5, "РК6", 3, 5, 3, 28),

        Group(6, "РК9", 3, 1, 3, 27),
        Group(7, "ФН12", 3, 2, 4, 21),
        Group(8, "ФН12", 3, 3, 4, 26),
        Group(9, "Э9ц", 5, 4, 5, 24)
    ]

    groups_faculties = [
        GroupFaculty(1, 1),
        GroupFaculty(2, 1),
        GroupFaculty(3, 1),
        GroupFaculty(4, 2),
        GroupFaculty(5, 3),
        GroupFaculty(6, 3),
        GroupFaculty(7, 4),
        GroupFaculty(8, 4),
        GroupFaculty(9, 5)
    ]

    task1(faculties, groups)
    task2(faculties, groups)
    task3(faculties, groups, groups_faculties)

if __name__ == "__main__":
    main()

```

Результат работы программы

```

Run  RK_1 PikYap x
C:\Users\AnnaA\AppData\Local\Programs\Python\Python311\
Запрос № 1
ИУ5ц-51 Информатика и системы управления
ИУ6ц-52 Информатика и системы управления
ИУ7-33 Информатика и системы управления

Запрос № 2
Энергомашиностроение 24
Фундаментальные науки 26
Инженерный бизнес и менеджмент 27
Робототехника и комплексная автоматизация 28
Информатика и системы управления 29

Запрос № 3
ИБМ7-34 Инженерный бизнес и менеджмент
ИУ5ц-51 Информатика и системы управления
ИУ6ц-52 Информатика и системы управления
ИУ7-33 Информатика и системы управления
РК6-35 Робототехника и комплексная автоматизация
РК9-31 Робототехника и комплексная автоматизация
ФН12-32 Фундаментальные науки
ФН12-33 Фундаментальные науки
Э9ц-54 Энергомашиностроение

```