

**Московский государственный технический
университет им. Н.Э. Баумана**

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

Курс «Парадигмы и конструкции языков программирования»
Отчет по рубежному контролю №1
Вар. В-28

Выполнил:
студент группы
ИУ5-34Б
Варданян Д.А.

Проверил:
преподаватель
каф. ИУ5
Нардид А.Н.

Москва, 2024 г.

Текст программы:

```
class Group:
```

```
    def __init__(self, id : int, code : str, students : int, cathedra_id : int):
        self.id = id
        self.code = code
        self.students = students
        self.cathedra_id = cathedra_id
```

```
class Cathedra:
```

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

```
class GroupCathedra:
```

```
    # Многое ко многим
```

```
    def __init__(self, cathedra_id : int, group_id : int):
        self.cathedra_id = cathedra_id
        self.group_id = group_id
```

```
'''Список кафедр'''
```

```
cathedras = [
    Cathedra(1, 'Системы обработки информации и управления'),
    Cathedra(2, 'Техническая физика'),
    Cathedra(3, 'Компьютерные системы и сети')
]
```

```
# Преподаватели
```

```
groups = [
    Group(1, 'ИУ5-34Б', 27, 1),
    Group(2, 'ИУ5-33Б', 28, 1),
    Group(3, 'ИУ5-31Б', 25, 1),
    Group(4, 'ФН4-32Б', 18, 2),
    Group(5, 'ИУ6-31Б', 16, 3),
]
```

```
group_cathedras = [
    GroupCathedra(1, 1),
    GroupCathedra(1, 2),
    GroupCathedra(1, 3),
    GroupCathedra(2, 4),
    GroupCathedra(3, 5)
]
```

```
def main():
```

```

# Создание кортежей 1:M
one_to_many = [(group.code, group.students, cathedra.name)
                for cathedra in cathedras
                for group in groups
                if group.cathedra_id == cathedra.id]

# Создание кортежей M:M
many_to_many_temp = [(cathedra.name, elem.cathedra_id, elem.group_id)
                      for cathedra in cathedras
                      for elem in group_cathedras
                      if cathedra.id == elem.cathedra_id]

many_to_many = [(group.code, group.students, cathedra_name)
                 for cathedra_name, cathedra_id, group_id in many_to_many_temp
                 for group in groups if group.id == group_id]

print('№1') # Группы начинающиеся в "И"
res1_temp = list(filter(lambda x: x[0].startswith('И'), one_to_many)) # "А" заменено на "И",
т.к. в списке нет групп, начинающихся на "А"
res1 = [(code, cathedra) for code, _, cathedra in res1_temp]
print(*[': '.join(ans) for ans in res1], sep='\n')

print('\n№2') # минимальное кол-во студентов
res2_unsorted = []
for cathedra in cathedras:
    res_2_temp = list(filter(lambda x: x[2] == cathedra.name, one_to_many))
    if len(res_2_temp) > 0:
        cathedra_studs = [studs for _, studs, _ in res_2_temp]
        cath_studs_min = min(cathedra_studs)
        res2_unsorted.append((cathedra.name, cath_studs_min))

res2 = sorted(res2_unsorted, key=lambda x: x[1])
print(*[': '.join(list(map(str, ans))) for ans in res2], sep='\n')

print('\n№3') # сортируем по названию группы
res_3_temp = sorted(many_to_many, key=lambda x: x[0])
res_3 = [(code, cath) for code, _, cath in res_3_temp]
print(*[': '.join(ans) for ans in res_3], sep='\n')

if __name__ == '__main__':
    main()

```

Вывод программы:

№1

ИУ5-34Б: Системы обработки информации и управления

ИУ5-33Б: Системы обработки информации и управления

ИУ5-31Б: Системы обработки информации и управления

ИУ6-31Б: Компьютерные системы и сети

№2

Компьютерные системы и сети: 16

Техническая физика: 18

Системы обработки информации и управления: 25

№3

ИУ5-31Б: Системы обработки информации и управления

ИУ5-33Б: Системы обработки информации и управления

ИУ5-34Б: Системы обработки информации и управления

ИУ6-31Б: Компьютерные системы и сети

ФН4-32Б: Техническая физика