

**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 No1  
Вар. В-29

Выполнил:

студент группы ИУ5-  
34Б

Волков Кирилл

Проверил:

преподаватель каф.  
ИУ5

Нардид А.Н.

Москва, 2024 г.

## Код программы:

```
class department:
    def __init__(self, id: int, name: str, salary: int, faculty_id: int):
        self.id = id
        self.name = name
        self.salary = salary
        self.faculty_id = faculty_id

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

class faculty_department:
    # Многое ко многим
    def __init__(self, department_id: int, faculty_id: int):
        self.department_id = department_id
        self.faculty_id = faculty_id

# Список факультетов
faculties = [
    faculty(1, 'ИУ'),
    faculty(2, 'МТ'),
    faculty(3, 'ФН'),
]

# Список кафедр
departments = [
    department(1, 'ИУ-5', 90000, 1),
    department(2, 'ИУ-7', 75000, 1),
    department(3, 'МТ-3', 60000, 2),
    department(4, 'ФН-4', 82000, 3),
]

# Связи многие-ко-многим
faculty_departments = [
    faculty_department(1, 1),
    faculty_department(2, 1),
    faculty_department(3, 2),
    faculty_department(4, 3),
]

def main():
    # Создание кортежей 1:M
    one_to_many = [(dep.name, dep.salary, fac.name)
                    for fac in faculties
                    for dep in departments
                    if dep.faculty_id == fac.id]

    # Создание кортежей M:M
    many_to_many_temp = [(fac.name, fd.department_id)
                           for fac in faculties
                           for fd in faculty_departments
                           if fac.id == fd.faculty_id]

    many_to_many = [(dep.name, dep.salary, fac_name)
                     for fac_name, department_id in many_to_many_temp]
```

```

        for dep in departments if dep.id == department_id]

print('№1') # Кафедры с названиями на "И"
res1_temp = list(filter(lambda x: x[0].startswith('И'), one_to_many))
res1 = [(name, faculty) for name, _, faculty in res1_temp]
print(*[': '.join(ans) for ans in res1], sep='\n')

print('№2') # Минимальная зарплата в каждом факультете
res2_unsorted = []
for fac in faculties:
    res_2_temp = list(filter(lambda x: x[2] == fac.name, one_to_many))
    if len(res_2_temp) > 0:
        salaries = [salary for _, salary, _ in res_2_temp]
        min_salary = min(salaries)
        res2_unsorted.append((fac.name, min_salary))

res2 = sorted(res2_unsorted, key=lambda x: x[1])
print(*[': '.join([name, f"{min_salary:.2f}"]) for name, min_salary in
res2], sep='\n')

print('№3') # Все связанные кафедры и факультеты
res3 = sorted(many_to_many, key=lambda x: x[0])
print(*[': '.join([department, faculty]) for department, _, faculty in
res3], sep='\n')

if __name__ == '__main__':
    main()

```

Вывод программы:

```

Run: main ×
/usr/local/bin/python3.9 /Users/volkov_kirill/Desktop/python_projects/Rk1/main.py
№1
ИУ-5: ИУ
ИУ-7: ИУ
№2
МТ: 60000.00
ИУ: 75000.00
ФН: 82000.00
№3
ИУ-5: ИУ
ИУ-7: ИУ
МТ-3: МТ
ФН-4: ФН

Process finished with exit code 0

```