

КОД ПРОГРАММЫ

Вариант Д29

```
class Department:
```

```
    def __init__(self, department_id, name, fee, faculty_id):
```

```
        self.department_id = department_id
```

```
        self.name = name
```

```
        self.fee = fee
```

```
        self.faculty_id = faculty_id
```

```
class Faculty:
```

```
    def __init__(self, faculty_id, name):
```

```
        self.faculty_id = faculty_id
```

```
        self.name = name
```

```
class DepartmentFaculty:
```

```
    def __init__(self, department_id, faculty_id):
```

```
        self.department_id = department_id
```

```
        self.faculty_id = faculty_id
```

```
def task(number: str) -> None:
```

```
    print(f"\n\nЗадание Д{number}:\n")
```

```
    return
```

```
def make_one_to_many(departments, faculties) -> list:
```

```
    one_to_many = [
```

```

        (department.name, department.fee, faculty.name)
    for faculty in faculties
    for department in departments
    if department.faculty_id == faculty.faculty_id
]

return one_to_many

```

def make_many_to_many(departments, faculties, departments_faculties) -> list:

```

    many_to_many_tmp = [
        (faculty.name, department_faculty.faculty_id,
        department_faculty.department_id)
        for faculty in faculties
        for department_faculty in departments_faculties
        if department_faculty.faculty_id == faculty.faculty_id
    ]

```

```

many_to_many = [
    (department.name, department.fee, faculty_name)
    for faculty_name, _, department_id in many_to_many_tmp
    for department in departments
    if department.department_id == department_id
]

```

```

return many_to_many

```

def task1(one_to_many) -> None: # Все заканчивающиеся на "2"

```
task("1")
result = list(filter(lambda x: x[0].endswith("2"), one_to_many))
for i in result:
    print(i[0], i[2])
return
```

```
def task2(one_to_many, faculties) -> None: # Средняя сумма по каждому факультету
```

```
    task("2")
    temp_list = []
    for i in faculties:
        temp_sum = 0
        temp_count = 0
        for j in one_to_many:
            if j[2] == i.name:
                temp_count += 1
                temp_sum += j[1]
        temp_list.append([i.name, temp_sum/temp_count])
    for name, average in sorted(temp_list, key=lambda x: x[1], reverse=True):
        print(name, average)
    return
```

```
def task3(many_to_many, faculties) -> None: # Все начинающиеся с "И"
```

```
    task("3")
    print("Факультеты:")
    for i in faculties:
        if i.name.startswith("И"):
```

```
        print(i.name)
print("Кафедры:")
for i in many_to_many:
    if i[2].startswith("И"):
        print(i[0])
return
```

def main() -> None:

```
    departments = [
        Department(1, "ИУ1", 100000, 1),
        Department(2, "ИУ2", 200000, 1),
        Department(3, "ИБМ1", 300000, 2),
        Department(4, "ИБМ2", 400000, 2),
        Department(5, "БМТ1", 500000, 3),
        Department(5, "БМТ2", 600000, 3)
    ]
```

```
    faculties = [
        Faculty(1, "Информатика и системы управления"),
        Faculty(2, "Инженерный бизнес и менеджмент"),
        Faculty(3, "Биомедицинская техника")
    ]
```

```
    departments_faculties = [
        DepartmentFaculty(1, 1),
        DepartmentFaculty(2, 1),
        DepartmentFaculty(3, 2),
        DepartmentFaculty(4, 2),
```

```

        DepartmentFaculty(5, 3),
        DepartmentFaculty(6, 3)
    ]

    one_to_many = make_one_to_many(departments, faculties)

    many_to_many = make_many_to_many(departments, faculties,
departments_faculties)

    task1(one_to_many)
    task2(one_to_many, faculties)
    task3(many_to_many, faculties)

    return

if __name__ == "__main__":
    main()

```

РЕЗУЛЬТАТ РАБОТЫ

Задание Д1:

ИУ2 Информатика и системы управления

ИБМ2 Инженерный бизнес и менеджмент

БМТ2 Биомедицинская техника

Задание Д2:

Биомедицинская техника 550000.0

Инженерный бизнес и менеджмент 350000.0

Информатика и системы управления 150000.0

Задание ДЗ:

Факультеты:

Информатика и системы управления

Инженерный бизнес и менеджмент

Кафедры:

ИУ1

ИУ2

ИБМ1

ИБМ2