

```
from operator import itemgetter
```

```
class Emp:
```

```
    """Сотрудник"""
```

```
    def __init__(self, id, fio, sal, dep_id):
```

```
        self.id = id
```

```
        self.fio = fio
```

```
        self.sal = sal
```

```
        self.dep_id = dep_id
```

```
class Dep:
```

```
    """Отдел"""
```

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

```
        self.id = id
```

```
        self.name = name
```

```
class EmpDep:
```

```
    """
```

```
    'Сотрудники отдела' для реализации
```

```
    связи многие-ко-многим
```

```
    """
```

```
    def __init__(self, dep_id, emp_id):
```

```
        self.dep_id = dep_id
```

```
        self.emp_id = emp_id
```

```
# Отделы
```

```
deps = [
```

```
    Dep(1, 'отдел кадров'),
```

```
    Dep(2, 'архивный отдел ресурсов'),
```

```
    Dep(3, 'бухгалтерия'),
```

```
    Dep(11, 'отдел (другой) кадров'),
```

```
    Dep(22, 'архивный (другой) отдел ресурсов'),
```

```
Dep(33, '(другая) бухгалтерия'),  
]
```

```
# Сотрудники
```

```
emps = [  
    Emp(1, 'Артамонов', 25000, 1),  
    Emp(2, 'Пронин', 35000, 2),  
    Emp(3, 'Осада', 45000, 3),  
    Emp(4, 'Папин', 35000, 3),  
    Emp(5, 'Заира', 25000, 3),  
]
```

```
emps_deps = [  
    EmpDep(1,1),  
    EmpDep(2,2),  
    EmpDep(3,3),  
    EmpDep(3,4),  
    EmpDep(3,5),  
    EmpDep(11,1),  
    EmpDep(22,2),  
    EmpDep(33,3),  
    EmpDep(33,4),  
    EmpDep(33,5),  
]
```

```
def main():
```

```
    """Основная функция"""
```

```
    # Задание 1: (Этот раздел не требует изменений)
```

```
    one_to_many = [(e.fio, e.sal, d.name) for d in deps for e in emps if e.dep_id == d.id]
```

```
    print('Задание A1')
```

```
    res_11 = sorted(one_to_many, key=itemgetter(2)) #Сортировка по имени отдела
```

```

print(res_11)

# Задание 2: (Исправленная логика суммирования зарплат)
print('\nЗадание A2')
dep_salaries = {}
for d in deps:
    total_salary = sum(e.sal for e in emps if e.dep_id == d.id)
    dep_salaries[d.name] = total_salary
res_12 = sorted(dep_salaries.items(), key=itemgetter(1), reverse=True)
print(res_12)

# Задание 3: (Улучшенная обработка many-to-many)
print('\nЗадание A3')
dep_emp_map = {}
for ed in emps_deps:
    dep = next((d for d in deps if d.id == ed.dep_id), None)
    emp = next((e for e in emps if e.id == ed.emp_id), None)
    if dep and emp:
        dep_name = dep.name
        emp_name = emp.fio
        if dep_name not in dep_emp_map:
            dep_emp_map[dep_name] = set()
        dep_emp_map[dep_name].add(emp_name)

res_13 = {dep: sorted(list(emps)) for dep, emps in dep_emp_map.items()}
print(res_13)

if __name__ == '__main__':
    main()

```

Результаты выполнения :

Задание А1

[('Пронин', 35000, 'архивный отдел ресурсов'), ('Осада', 45000, 'бухгалтерия'), ('Папин', 35000, 'бухгалтерия'), ('Заира', 25000, 'бухгалтерия'), ('Артамонов', 25000, 'отдел кадров')]

Задание А2

[('бухгалтерия', 105000), ('архивный отдел ресурсов', 35000), ('отдел кадров', 25000), ('отдел (другой) кадров', 0), ('архивный (другой) отдел ресурсов', 0), ('(другая) бухгалтерия', 0)]

Задание А3

{'отдел кадров': ['Артамонов'], 'архивный отдел ресурсов': ['Пронин'], 'бухгалтерия': ['Заира', 'Осада', 'Папин'], 'отдел (другой) кадров': ['Артамонов'], 'архивный (другой) отдел ресурсов': ['Пронин'], '(другая) бухгалтерия': ['Заира', 'Осада', 'Папин']}

PS C:\Users\Evgen\OneDrive\Рабочий стол>