

Текст main.py

```
from operator import itemgetter

class Detail:
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Supplier:
    def __init__(self, id, name, salary, detail_id):
        self.id = id
        self.name = name
        self.salary = salary
        self.detail_id = detail_id

class SupplierDetail:
    def __init__(self, supplier_id, detail_id):
        self.supplier_id = supplier_id
        self.detail_id = detail_id

details = [
    Detail(1, "Труба"),
    Detail(2, "Деталь 2"),
    Detail(3, "Сложная деталь"),
    Detail(4, "Простая деталь"),
]

suppliers = [
    Supplier(1, "Поставщик А", 5000, 1),
    Supplier(2, "Поставщик Б", 3000, 2),
    Supplier(3, "Поставщик В", 4000, 3),
    Supplier(4, "Поставщик Г", 2000, 1),
    Supplier(5, "Поставщик Д", 2500, 4),
]

supplier_details = [
    SupplierDetail(1, 1),
    SupplierDetail(2, 2),
    SupplierDetail(3, 3),
    SupplierDetail(4, 1),
    SupplierDetail(5, 4),
]

# СВЯЗЬ ОДИН-КО-МНОГИМ
def task1():
    result = [(detail.name, supplier.name)
               for detail in details
               for supplier in suppliers
               if supplier.detail_id == detail.id]
    return sorted(result, key=itemgetter(0))
```

```

# Суммарная зарплата
def task2():
    result = []
    for detail in details:
        total_salary = sum(supplier.salary for supplier in suppliers if
supplier.detail_id == detail.id)
        result.append((detail.name, total_salary))
    return sorted(result, key=itemgetter(1), reverse=True)

# Связь многие-ко-многим
def task3():
    result = []
    for detail in details:
        if "деталь" in detail.name.lower():
            suppliers_in_detail = [supplier.name
for supplier in suppliers
for sd in supplier_details
if sd.detail_id == detail.id and
sd.supplier_id == supplier.id]
            result.append((detail.name, suppliers_in_detail))
    return result

# Main function
def main():
    print("Задание 1: Список связанных поставщиков и деталей, отсортированный
по деталям:")
    print(task1())

    print("\nЗадание 2: Список деталей с суммарной зарплатой поставщиков,
отсортированный по суммарной зарплате:")
    print(task2())

    print("\nЗадание 3: Список деталей, содержащих слово 'деталь', и
работающих в них поставщиков:")
    print(task3())

if __name__ == '__main__':
    main()

```

Результат main.py

```

Задание 1: Список связанных поставщиков и деталей, отсортированный по деталям:
[('Деталь 2', 'Поставщик Б'), ('Простая деталь', 'Поставщик Д'), ('Сложная деталь', 'Поставщик В'), ('Труба', 'Поставщик А'), ('Труба', 'Поставщик Г')]

Задание 2: Список деталей с суммарной зарплатой поставщиков, отсортированный по суммарной зарплате:
[('Труба', 7000), ('Сложная деталь', 4000), ('Деталь 2', 3000), ('Простая деталь', 2500)]

Задание 3: Список деталей, содержащих слово 'деталь', и работающих в них поставщиков:
[('Деталь 2', ['Поставщик Б']), ('Сложная деталь', ['Поставщик В']), ('Простая деталь', ['Поставщик Д'])]

Process finished with exit code 0

```