Текст программы:

```
class Detail:
    def __init__(self, id, weight, type, supplier_id):
        self.id = id
        self.weight = weight
        self.type = type
        self.supplier_id = supplier_id

class Supplier:
    def __init__(self, id, name):
        self.id = id
        self.name = name
    def __lt__(self, obj):
        return (self.name<obj.name)

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

# данные

list_of_details = [Detail(1, 50, "screw", 3), Detail(2, 300, "casing", 4),
                   Detail(3, 60, "bolt", 3), Detail(4, 178, "cable holder", 1),
                   Detail(5, 312, "casing", 2), Detail(6, 100, "wire", 1),
                   Detail(7, 30, "nut", 3)]
list_of_suppliers= [Supplier(1, "Wires and Sons"), Supplier(2, "Another Firm co"),
                    Supplier(3, "Connectors Inc."), Supplier(4, "Casings Production")]
connecting_list = [DetSup(1, 3), DetSup(2, 4), DetSup(3, 3), DetSup(4, 1), DetSup(7,
2),
                   DetSup(1, 2), DetSup(5, 2), DetSup(6, 1), DetSup(7, 3), DetSup(2, 2)]

def unit_test(**kwargs):
    def inner_decorator(func):
        def wrapped(*args):
            assert 'input' in kwargs and 'output' in kwargs
            for i in range(len(kwargs['input'])):
                response = func(*kwargs['input'][i])
                assert response == kwargs['output'][i], f'Incorrect return of function
{func.__name__}({kwargs["input"][i]}) - {response}. Must be {kwargs["output"][i]}'
            print(f"Test function {func.__name__} complete. Not found errors")
            return func(*args)
        return wrapped
    return inner_decorator
```

```
@unit_test(input=[], output=[
        {'Connectors Inc.': ['screw', 'bolt', 'nut'], 'Casings Production': ['casing']}
    ])
def f1():
        res1 = {}
        for sup in list_of_suppliers:
            if (sup.name.startswith('C')):
                res1[sup.name] = [det.type for det in list_of_details if det.supplier_id ==
sup.id]
        print(res1)

@unit_test(input=[], output=[
        [('Another Firm co', ('casing', 312)), ('Casings Production', ('casing', 300)),
('Wires and Sons', ('cable holder', 178)), ('Connectors Inc.', ('bolt', 60))]
    ])
def f2():
    res2 = {}
    for sup in list_of_suppliers:
        dets = [det for det in list_of_details if det.supplier_id == sup.id]
        if(len(dets)!=0):
            max_det = max(dets, key=lambda d: d.weight)
            res2[sup.name] = (max_det.type, max_det.weight)
    print(sorted(res2.items(), key=lambda d: d[1][1], reverse=True))

@unit_test(input=[], output=[{'Another Firm co': [('nut', 30, 'Another Firm co'),
('screw', 50, 'Another Firm co'), ('casing', 312, 'Another Firm co'), ('casing', 300,
'Another Firm co')], 'Casings Production': [('casing', 300, 'Casings Production')],
'Connectors Inc.': [('screw', 50, 'Connectors Inc.'), ('bolt', 60, 'Connectors Inc.'), ('nut',
30, 'Connectors Inc.')], 'Wires and Sons': [('cable holder', 178, 'Wires and Sons'),
('wire', 100, 'Wires and Sons')]}
                            ])
def f3():
    res3 = {}
    many_to_many_temp = [(s.name, sd.supplier_id, sd.detail_id)
        for s in list_of_suppliers
        for sd in connecting_list
        if s.id==sd.supplier_id]
    many_to_many = [(d.type, d.weight, name)
        for name, supplier_id, detail_id in many_to_many_temp
        for d in list_of_details if d.id==detail_id]
    # список поставщиков сортируется по алфавиту, в классе реализовано
сравнение
    for sup in sorted(list_of_suppliers):
        res3[sup.name] = list(filter(lambda i: i[2] == sup.name, many_to_many))
```

```
    print(res3)

def main():
    print("Задание 1")
    f1()
    print("Задание 2")
    f2()
    print("Задание 3")
    f3()

if __name__ == '__main__':
    main()
```

Вывод:

```
Задание 1
Test function f1 complete. Not found errors
{'Connectors Inc.': ['screw', 'bolt', 'nut'], 'Casings Production': ['casing']}
Задание 2
Test function f2 complete. Not found errors
[('Another Firm co', ('casing', 312)), ('Casings Production', ('casing', 300)), ('Wires and Sons', ('cable holder', 178)), ('Connectors Inc.', ('bolt', 60))]
Задание 3
Test function f3 complete. Not found errors
{'Another Firm co': [('nut', 30, 'Another Firm co'), ('screw', 50, 'Another Firm co'), ('casing', 312, 'Another Firm co'), ('casing', 300, 'Another Firm co')], 'Casings Production': [('casing', 300, 'Casings Production')], 'Connectors Inc.': [('screw', 50, 'Connectors Inc.'), ('bolt', 60, 'Connectors Inc.'), ('nut', 30, 'Connectors Inc.')], 'Wires and Sons': [('cable holder', 178, 'Wires and Sons'), ('wire', 100, 'Wires and Sons')]}
```