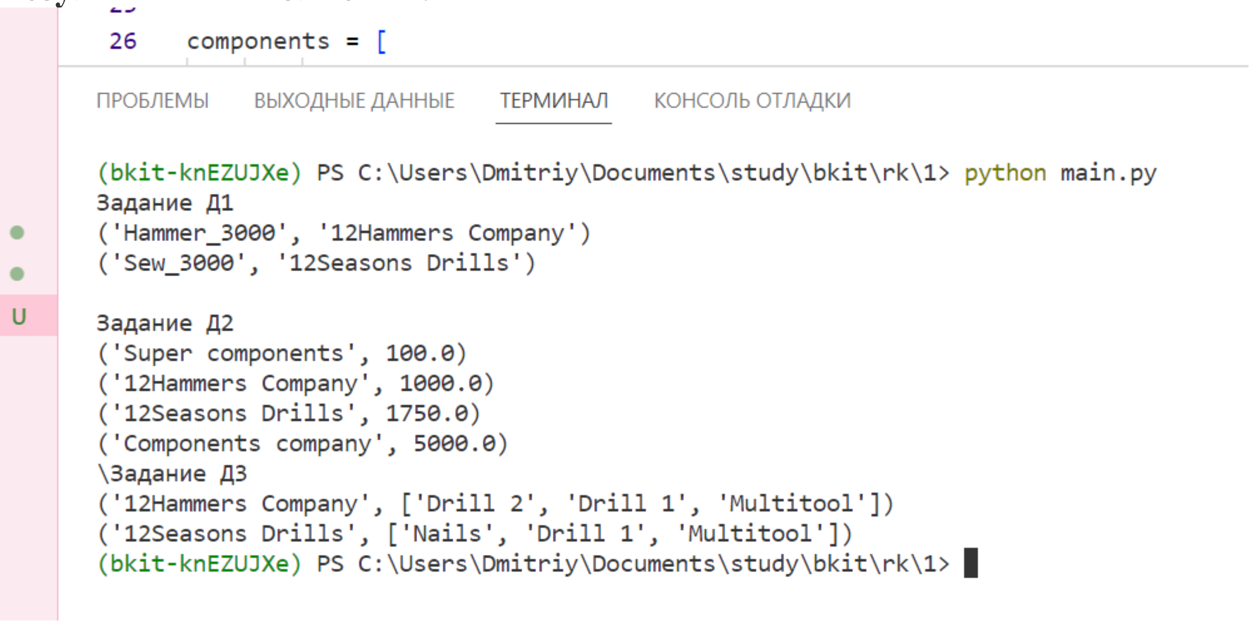
Текст программы:

```
class Component:
    def __init__(self, id, name, cost, provider_id):
        self.id = id
        self.name = name
        self.cost = cost
        self.provider_id = provider_id

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

class Providers_components:
    def __init__(self, component_id, provider_id):
        self.component_id = component_id
        self.provider_id = provider_id


providers = [
             Provider(1, '12Hammers Company'),
             Provider(2, 'Super components'),
             Provider(3, 'Components company'),
             Provider(4, '12Seasons Drills')
]

components = [
            Component(1, 'Hammer_3000', 1000, 1),
            Component(2, 'Nails', 100, 2),
            Component(3, 'Drill 1', 5000, 3),
            Component(4, 'Drill 2', 5000, 3),
            Component(5, 'Sew_3000', 3000, 4),
            Component(6, 'Multitool', 500, 4)
]

providers_components = [
                Providers_components(2, 2),
                Providers_components(4, 1),
                Providers_components(1, 3),
                Providers_components(5, 2),
                Providers_components(2, 4),
                Providers_components(2, 2),
                Providers_components(3, 1),
                Providers_components(3, 4),
                Providers_components(6, 4),
                Providers_components(6, 1),
]

def list_average(lst):
    if not(len(lst)): return 0
    else: return sum(lst)/len(lst)

def main():
    '''соотношения один-ко-многим и многие-ко-многим'''
    one_to_many = [(c.name, c.cost, p.name)
        for p in providers
        for c in components
        if c.provider_id==p.id]

    many_to_many_temp = [(p.name, pc.provider_id, pc.component_id)
        for p in providers
        for pc in providers_components
        if p.id == pc.provider_id]

    many_to_many = [(c.name, c.cost, provider_name)
        for provider_name, provider_id, component_id in many_to_many_temp
        for c in components if c.id == component_id]

    print('Задание Д1')

    '''
    список деталей (с указанием производителя),
    с назаванием, кончающимся на "3000":
    '''

    print(*[(x[0], x[2]) for x in one_to_many if x[0][-4:] == "3000"], sep='\n')

    print('\nЗадание Д2')

    '''
    список производителей,
    отсортированный по средней стоимости производимых ими деталей (с указанием
средней цены)
    '''

    providers_avg_cost = []

    for p in providers:
        p_components_list = [x[1] for x in one_to_many if x[2] == p.name]
        providers_avg_cost.append((p.name, list_average(p_components_list)))

    print(*sorted(providers_avg_cost, key = lambda tup: tup[1]), sep='\n')

    print('\Задание Д3')

    '''
    товары, у которых
    название производителя начинается на "12":
    '''

    names = (p.name for p in providers if p.name[:2] == "12")
    print(*[(name, [x[0] for x in many_to_many if x[2] == name]) for name in
names], sep='\n')


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

Результаты выполнения:



```
(bkit-knEZUJXe) PS C:\Users\Dmitriy\Documents\study\bkit\rk\1> python main.py
Задание Д1
('Hammer_3000', '12Hammers Company')
('Sew_3000', '12Seasons Drills')

Задание Д2
('Super components', 100.0)
('12Hammers Company', 1000.0)
('12Seasons Drills', 1750.0)
('Components company', 5000.0)
\Задание Д3
('12Hammers Company', ['Drill 2', 'Drill 1', 'Multitool'])
('12Seasons Drills', ['Nails', 'Drill 1', 'Multitool'])
(bkit-knEZUJXe) PS C:\Users\Dmitriy\Documents\study\bkit\rk\1>
```