

Текст изменённой для модульного тестирования программы из РК1:

Файл tasks.py

```
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)

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]

def task_1():
    return [(x[0], x[2]) for x in one_to_many if x[0][-4:] == "3000"]

def task_2():
    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)))

    return sorted(providers_avg_cost, key = lambda tup: tup[1])

def task_3():
    names = (p.name for p in providers if p.name[:2] == "12")
    return [(name, [x[0] for x in many_to_many if x[2] == name]) for name in
names]

```

Текст программы для модульного тестирования:

Файл tdd.py

```

import unittest
from tasks import task_1, task_2, task_3

task_1_result = [('Hammer_3000', '12Hammers Company'), ('Sew_3000', '12Seasons Drills')]
task_2_result = [ ('Super components', 100.0), ('12Hammers Company', 1000.0),
                  ('12Seasons Drills', 1750.0), ('Components company', 5000.0)]
task_3_result = [('12Hammers Company', ['Drill 2', 'Drill 1', 'Multitool']),
                  ('12Seasons Drills', ['Nails', 'Drill 1', 'Multitool'])]

class TasksTestCase(unittest.TestCase):
    def test_task_1(self):
        self.assertEqual(task_1_result, task_1())

```

```

def test_task_2(self):
    self.assertEqual(task_2_result, task_2())
def test_task_3(self):
    self.assertEqual(task_3_result, task_3())

```

Результаты выполнения модульного тестирования:

```

PS C:\Users\Dmitriy\Documents\study\BKIT\rk\2> pytest -v tdd.py
===== test session starts =====
platform win32 -- Python 3.9.13, pytest-7.2.0, pluggy-1.0.0 -- C:\Users\Dmitriy\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundat
ion.Python.3.9_qbz5n2kfra8p0\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Dmitriy\Documents\study\BKIT\rk\2
plugins: bdd-6.1.1
collected 3 items

tdd.py::TasksTestCase::test_task_1 PASSED [ 33%]
tdd.py::TasksTestCase::test_task_2 PASSED [ 66%]
tdd.py::TasksTestCase::test_task_3 PASSED [100%]

===== 3 passed in 0.04s =====
PS C:\Users\Dmitriy\Documents\study\BKIT\rk\2> █

```