

Рубежный контроль №2 по дисциплине «Базовые компоненты интернет-технологий»
Вариант 19Б

Файл Detail.py

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

Файл Manufacturer.py

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

Файл DetMan.py

```
class DetMan:
    def __init__(self, detailID, manID):
        self.detailID = detailID
        self.manID = manID
```

Файл main.py

```
from operator import itemgetter

from DetMan import DetMan
from Detail import Detail
from Manufacturer import Manufacturer

# Производители
manufacturers = [
    Manufacturer(1, 'Lmi'),
    Manufacturer(2, 'Lucas'),
    Manufacturer(3, 'Hella'),
    Manufacturer(11, 'не Lmi'),
    Manufacturer(22, 'не Lucas'),
    Manufacturer(33, 'не Hella')
]

# Детали
details = [
    Detail(1, "ny0jsa", 100, 1),
    Detail(2, "hainfb", 213, 2),
    Detail(3, "shcbe", 123, 2),
    Detail(4, "dnufv", 293, 3),
    Detail(5, "djifb", 90, 3)
]

detsMans = [
    DetMan(1, 1),
    DetMan(2, 2),
    DetMan(2, 3),
    DetMan(3, 4),
    DetMan(3, 5),

    DetMan(11, 1),
    DetMan(22, 2),
    DetMan(22, 3),
    DetMan(33, 4),
    DetMan(33, 5),
]

def task1(oneToMany):
    res_1 = sorted(oneToMany, key=itemgetter(0))
    print('Задание 61')
    for row in res_1:
        print(row[:-1])
    return res_1

def task2(oneToMany):
    res_2 = []
    for i in range(3):
        count_details = list(filter(lambda j: j[2] == manufacturers[i].name, oneToMany))
        res_2.append((manufacturers[i].name, len(count_details)))
    res_2 = sorted(res_2, key=itemgetter(1), reverse=True)
    print('Задание 62')
    for row in res_2:
        print(row)
    return res_2

def task3(manyToMany):
    res_3 = []
    for i in manyToMany:
        if i[0][-2:] == 'fb':
            res_3.append(i)
    print('Задание 63')
    for row in res_3:
        print(row)
    return res_3

def main():
    # один ко многим
    oneToMany = [(d.name, d.price, m.name)
                  for d in details
                  for m in manufacturers
                  if d.manID == m.id]

    # многие ко многим
    manyToManyTemp = [(m.name, dm.detailID, dm.manID)
                      for m in manufacturers
                      for dm in detsMans
                      if m.id == dm.manID]
    manyToMany = [(d.name, d.price, man_name)
                  for man_name, manID, detailID in manyToManyTemp
                  for d in details if d.id == detailID]

    task1(oneToMany)
    task2(oneToMany)
    task3(manyToMany)
```

Файл test.py

```
from unittest import TestCase, main
from main import task1, task2, task3

class testrk2(TestCase):
    def test_task1(self):
        self.assertEqual(task1(
            [('djifb', 90, 'Hella'),
             ('dnufv', 293, 'Hella'),
             ('hainfb', 213, 'Lucas'),
             ('ny0jsa', 100, 'Lmi'),
             ('shcbe', 123, 'Lucas')]),
            [('djifb', 90, 'Hella'),
             ('dnufv', 293, 'Hella'),
             ('hainfb', 213, 'Lucas'),
             ('ny0jsa', 100, 'Lmi'),
             ('shcbe', 123, 'Lucas')]))

    def test_task2(self):
        assert len(task2([('djifb', 90, 'Hella'),
                          ('dnufv', 293, 'Hella'),
                          ('hainfb', 213, 'Lucas'),
                          ('ny0jsa', 100, 'Lmi'),
                          ('shcbe', 123, 'Lucas')])) == 3

    def test_task3(self):
        self.assertEqual(task3(
            [('djifb', 90, 'Hella'),
             ('dnufv', 293, 'Hella'),
             ('hainfb', 213, 'Lucas'),
             ('ny0jsa', 100, 'Lmi'),
             ('shcbe', 123, 'Lucas')]),
            [('djifb', 90, 'Hella'),
             ('hainfb', 213, 'Lucas')]))
```

Результат работы тестов

✓ Tests passed: 3 of 3 tests – 6 ms

Ran 3 tests in 0.009s

OK