



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования "Московский государственный технический университет
им. Н. Э. Баумана (национальный исследовательский университет)"

Факультет "Информатика и системы управления"
Кафедра ИУ5 "Системы обработки информации и управления"

Рубежный контроль №2

По дисциплине "Базовые компоненты интернет-технологий"

Выполнил:
студент группы ИУ5-35Б
Чернецов С.А.

Москва, 2021 г.

Постановка задачи:

(Вариант запросов - Д, вариант предметной области - 19)

1. «Деталь» и «Производитель» связаны соотношением один ко многим. Выведите список всех деталей, у которых название заканчивается на «т», и названия их производителей.
2. «Деталь» и «Производитель» связаны соотношением один ко многим. Выведите список производителей со средним весом детали у каждого производителя, отсортированный по среднему весу.
3. «Деталь» и «Производитель» связаны соотношением многие ко многим. Вывести список всех производителей, у которых название начинается на G, и список производимых ими деталей.
4. Провести рефакторинг программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
5. Для текста программы рубежного контроля №1 создать модульные тесты с применением TDD - фреймворка (3 теста).

Листинг

rk1.py

```
# -*- coding utf-8 -*-
# используется для сортировки

from operator import itemgetter
class Detail:
    def __init__(self, id, name, weight, man_id):
        self.id = id
        self.name = name
        self.weight = weight
        self.man_id = man_id

class Manufacturer:
    def __init__(self, id, name):
        self.id = id
        self.name = name
class DetailManufacturer:
    def __init__(self, det_id, man_id):
        self.det_id = det_id
        self.man_id = man_id

# Отделы
details = [
    Detail(1, 'гайка', 50, 1),
    Detail(2, 'болт', 35, 2),
    Detail(3, 'шайба', 15, 2),
    Detail(4, 'вал', 800, 2),
    Detail(5, 'ремень', 250, 1),
    Detail(6, 'шестерня', 650, 1),
    Detail(7, 'заклёпка', 110, 2),
    Detail(8, 'винт', 60, 1)
]

# Сотрудники
manufacturers = [
    Manufacturer(1, 'Siemens'),
    Manufacturer(2, 'Mann'),
```

```

    Manufacturer(3, 'Gabriel'),
    Manufacturer(4, 'Gates')
]
det_man = [
    DetailManufacturer(1, 1),
    DetailManufacturer(2, 2),
    DetailManufacturer(3, 2),
    DetailManufacturer(4, 2),
    DetailManufacturer(5, 1),
    DetailManufacturer(6, 1),
    DetailManufacturer(7, 2),
    DetailManufacturer(8, 2),
]

class Test:
    def __init__(self, details, manufacturers, det_man):
        self.details = details
        self.manufacturers = manufacturers
        self.det_man=det_man

        self.one_to_many = [(d.name,d.weight, m.name)
                             for d in details
                             for m in manufacturers
                             if d.man_id ==m.id ]

        self.many_to_many_temp = [(m.name, ed.man_id, ed.det_id)
                                   for m in details
                                   for ed in det_man
                                   if m.id == ed.det_id]

        self.many_to_many = [(d.name, d.weight, man_name)
                              for man_name, man_id, det_id in self.many_to_many_temp
                              for d in details if d.id == det_id]

    def Zadanie1(self):
        one_to_many = [(d.name, d.weight, m.name)
                       for d in details
                       for m in manufacturers
                       if d.man_id == m.id]

        print('Задание Д1')
        res1=[]
        for name, man_id, man_name in self.one_to_many:
            if name.endswith('т'):
                res1.append((name, man_name))
        print(res1)
        return res1

    def Zadanie2 (self):
        print('\nЗадание Д2')
        res2_unsorted = []
        # Перебираем всех производителей
        for m in manufacturers:
            # Список деталей
            m_details = list(filter(lambda i: i[2] == m.name, self.one_to_many))
            # Если производитель не пустой
            if len(m_details) > 0:
                # Вес детали
                d_weights = [weight for _, weight, _ in m_details]
                # Суммарный вес деталей
                d_weights_sum = sum(d_weights)
                average_weight = d_weights_sum/len(d_weights)
                res2_unsorted.append((m.name, average_weight))

        # Сортировка по среднему весу
        res2 = sorted(res2_unsorted, key=itemgetter(1), reverse=True)
        print(res2)
        return res2

    def Zadanie3(self):

```

```

print('\nЗадание Д3')

res3 = {}
#Перебираем всех производителей
for m in manufacturers:
    if 'G' in m.name:
        # Список производителей
        m_details = list(filter(lambda i: i[2] == m.name,
self.many_to_many))
        m_details_names = [x for x,_,_ in m_details]
        # Добавляем результат в словарь
        # ключ - производитель, значение - список деталей
        res3[m.name] = m_details_names
print(res3)
return res3

if __name__ == '__main__':
    a = Test(details, manufacturers, det_man)
    a.Zadanie1()
    a.Zadanie2()
    a.Zadanie3()

```

Test.py

```

# -*- coding utf-8 -*-
import unittest

from rkl import Test, details, manufacturers, det_man

class Tester(unittest.TestCase):
    def test_Zadanie1(self):
        a = Test(details, manufacturers, det_man)
        t1 = [('БОЛТ', 'Mann'), ('ВИНТ', 'Siemens')]
        self.assertEqual(t1, a.Zadanie1())
    def test_Zadanie2(self):
        a = Test(details, manufacturers, det_man)
        t2 = [('Siemens', 252.5), ('Mann', 240.0)]
        self.assertEqual(t2, a.Zadanie2())
    def test_Zadanie3(self):
        a = Test(details, manufacturers, det_man)
        t3 = {'Gabriel': [], 'Gates': []}
        self.assertEqual(t3, a.Zadanie3())

if __name__ == '__main__':
    unittest.main()

```

Анализ результатов:

```
✓ Tests passed: 3 of 3 tests - 0 ms

C:\Users\Семён\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users\Семён\PyCharm Community Edition 2021.3\plugins\python-ce\helpers
Testing started at 21:38 ...
Launching unittests with arguments python -m unittest C:/Users/Семён/PycharmProjects/pythonProject/Test.py in C:/Users/Семён/PycharmProjects\p

Задание Д1
[('Болт', 'Mann'), ('винт', 'Siemens')]

Задание Д2
[('Siemens', 252.5), ('Mann', 240.0)]

Задание Д3
{'Gabriel': [], 'Gates': []}

Ran 3 tests in 0.010s

OK

Process finished with exit code 0
```