



**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчёт по рубежному контролю №2

Выполнил:

студент группы ИУ5-31Б
Фонин Максим Алексеевич

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

Описание задания

(Вариант предметной области - 20, вариант запросов - А)

1. Провести рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
2. Для текста программы рубежного контроля №1 создать модульные тесты с применением TDD - фреймворка (3 теста).

Текст программы

```
# main.py
# Вариант 20: Деталь - Поставщик
from operator import itemgetter

class Detail:
    """Деталь"""
    def __init__(self, id, name, mat, prov_id):
        self.id = id
        self.name = name
        self.mat = mat
        self.prov_id = prov_id

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

class DetProv:
    """
    'Поставщики деталей' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """
    def __init__(self, det_id, prov_id):
        self.det_id = det_id
        self.prov_id = prov_id

def task1(providers, details):
    one_to_many = [(d.name, d.mat, p.name)
                    for p in providers
                    for d in details
                    if d.prov_id == p.id]
    return sorted(one_to_many, key=itemgetter(0))

def task2(providers, details):
    res = list()

    one_to_many = [(d.name, d.mat, p.name)
                    for p in providers
                    for d in details
                    if d.prov_id == p.id]
    for p in providers:
        # Список деталей, поставляемых этим поставщиком
        p_dets = list(filter(lambda i: i[2]==p.name, one_to_many))
        if len(p_dets) > 0:
            # Поставщик и количество деталей, им поставляемых
            res.append((p.name, len(p_dets)))

    # Сортировка по количеству поставляемых деталей
    return sorted(res, key=itemgetter(1), reverse=True)
```

```

def task3(providers, details, dets_provs):
    res = dict()

    many_to_many = [(d.name, d.mat, p.name)
                     for d in details
                     for p in providers
                     for dp in dets_provs
                     if d.id == dp.det_id and
                        p.id == dp.prov_id]
    for p in providers:
        if p.name.startswith('ПК'):
            # Список деталей
            p_dets = list(filter(lambda i: i[2]==p.name, many_to_many))
            p_dets_names = [i for i,_,_ in p_dets]
            # Добавляем результат в словарь
            # ключ - поставщик, значение - список названий деталей
            res[p.name] = p_dets_names
    return res

def main():
    """Основная функция"""

    # Поставщики
    providers = [
        Provider(1, 'ПАО "Ижсталь"',),
        Provider(2, 'ПК "ГМК Норникель"',),
        Provider(3, 'ПК "Северсталь"',),
        Provider(4, 'ПК "РУСАЛ Красноярск"',),
        Provider(5, 'ООО "ПФ Инзенский ДФЗ"',),
        Provider(6, 'Тайшетский АЗ',),
        Provider(7, 'ПК Челябинский МК',),
        Provider(8, 'ПК Металлогорский МК')
    ]

    # Детали
    details = [
        Detail(1, 'Болт М2', 'Сталь', 1),
        Detail(2, 'Труба Д16Т', 'Никель', 2),
        Detail(3, 'Доска 160x20', 'Дерево', 5),
        Detail(4, 'Штуцер М26xШ22', 'Сталь', 3),
        Detail(5, 'Профиль 20x20x2000', 'Алюминий', 4),
        Detail(6, 'Фланец Д25', 'Алюминий', 8),
        Detail(7, 'Калиброванный прокат', 'Сталь', 1),
        Detail(8, 'Лента', 'Сталь', 1),
        Detail(9, 'Катодный лист Н-1У', 'Никель', 2),
        Detail(10, 'Фольга', 'Алюминий', 3),
        Detail(11, 'Колесный диск', 'Алюминий', 3),
        Detail(12, 'Катанка', 'Алюминий', 3)
    ]

    dets_provs = [
        DetProv(1, 1),
        DetProv(1, 3),
        DetProv(1, 7),
        DetProv(1, 8),

        DetProv(2, 2),

        DetProv(3, 5),

        DetProv(4, 1),
        DetProv(4, 3),
        DetProv(4, 7),
        DetProv(4, 8),

        DetProv(5, 4),
        DetProv(5, 6),
    ]

```

```

        DetProv(6, 4),
        DetProv(6, 6),

        DetProv(7, 1),
        DetProv(7, 3),
        DetProv(7, 7),
        DetProv(7, 8),

        DetProv(8, 1),
        DetProv(8, 3),
        DetProv(8, 7),
        DetProv(8, 8),

        DetProv(9, 2),

        DetProv(10, 4),
        DetProv(10, 6),

        DetProv(11, 4),
        DetProv(11, 6),

        DetProv(12, 4),
        DetProv(12, 6)
    ]

    print('Задание A1')
    res_1 = task1(providers, details)
    [print(res) for res in res_1]

    print('\nЗадание A2')
    res_2 = task2(providers, details)
    [print(res) for res in res_2]

    print('\nЗадание A3')
    res_3 = task3(providers, details, dets_provs)
    [print(key, value) for key, value in res_3.items()]

if __name__ == '__main__':
    main()

# tests.py
import unittest
from main import *

class Test(unittest.TestCase):
    def setUp(self):
        self.providers = [
            Provider(1, 'ПАО "Ижсталь"',),
            Provider(2, 'ПК "ГМК Норникель"',),
            Provider(3, 'ПК "Северсталь"',),
            Provider(4, 'ПК "РУСАЛ Красноярск"',),
            Provider(5, 'ООО "ПФ Инзенский ДФЗ"',),
            Provider(6, 'Тайшетский АЗ'),
            Provider(7, 'ПК Челябинский МК'),
            Provider(8, 'ПК Металлогорский МК')
        ]

        self.details = [
            Detail(1, 'Болт М2', 'Сталь', 1),
            Detail(2, 'Труба Д16Т', 'Никель', 2),
            Detail(3, 'Доска 160x20', 'Дерево', 5),
            Detail(4, 'Штуцер М26xШ22', 'Сталь', 3),
            Detail(5, 'Профиль 20x20x2000', 'Алюминий', 4),

```

```

Detail(6, 'Фланец Д25', 'Алюминий', 8),
Detail(7, 'Калиброванный прокат', 'Сталь', 1),
Detail(8, 'Лента', 'Сталь', 1),
Detail(9, 'Катодный лист Н-1У', 'Никель', 2),
Detail(10, 'Фольга', 'Алюминий', 3),
Detail(11, 'Колесный диск', 'Алюминий', 3),
Detail(12, 'Катанка', 'Алюминий', 3)
]

```

```

self.dets_provs = [
    DetProv(1, 1),
    DetProv(1, 3),
    DetProv(1, 7),
    DetProv(1, 8),

    DetProv(2, 2),

    DetProv(3, 5),

    DetProv(4, 1),
    DetProv(4, 3),
    DetProv(4, 7),
    DetProv(4, 8),

    DetProv(5, 4),
    DetProv(5, 6),

    DetProv(6, 4),
    DetProv(6, 6),

    DetProv(7, 1),
    DetProv(7, 3),
    DetProv(7, 7),
    DetProv(7, 8),

    DetProv(8, 1),
    DetProv(8, 3),
    DetProv(8, 7),
    DetProv(8, 8),

    DetProv(9, 2),

    DetProv(10, 4),
    DetProv(10, 6),

    DetProv(11, 4),
    DetProv(11, 6),

    DetProv(12, 4),
    DetProv(12, 6)
]

```

```

def test_task_1(self):

```

```

    key_fot_test = [
        ('Болт М2', 'Сталь', 'ПАО "Ижсталь"'),
        ('Доска 160x20', 'Дерево', 'ООО "ПФ Инзенский ДФЗ"'),
        ('Калиброванный прокат', 'Сталь', 'ПАО "Ижсталь"'),
        ('Катанка', 'Алюминий', 'ПК "Северсталь"'),
        ('Катодный лист Н-1У', 'Никель', 'ПК "ГМК Норникель"'),
        ('Колесный диск', 'Алюминий', 'ПК "Северсталь"'),
        ('Лента', 'Сталь', 'ПАО "Ижсталь"'),
        ('Профиль 20x20x2000', 'Алюминий', 'ПК "РУСАЛ Красноярск"'),
        ('Труба Д16Т', 'Никель', 'ПК "ГМК Норникель"'),
        ('Фланец Д25', 'Алюминий', 'ПК "Металлогорский МК"'),
        ('Фольга', 'Алюминий', 'ПК "Северсталь"'),
        ('Штуцер М26xШ22', 'Сталь', 'ПК "Северсталь"'),
    ]

```

```

]

self.assertEqual(task1(self.providers, self.details), key_fot_test)

def test_task_2(self):

    key_fot_test = [
        ('ПК "Северсталь"', 4),
        ('ПАО "Ижсталь"', 3),
        ('ПК "ГМК Норникель"', 2),
        ('ПК "РУСАЛ Красноярск"', 1),
        ('ООО "ПФ Инзенский ДФЗ"', 1),
        ('ПК Металлогорский МК', 1),
    ]

    self.assertEqual(task2(self.providers, self.details), key_fot_test)

def test_task_3(self):

    key_fot_test = {
        'ПК "ГМК Норникель"': ['Труба Д16Т', 'Катодный лист Н-1У'],
        'ПК "Северсталь"': ['Болт М2', 'Штуцер М26хШ22', 'Калиброванный
прокат', 'Лента'],
        'ПК "РУСАЛ Красноярск"': ['Профиль 20х20х2000', 'Фланец Д25',
'Фольга', 'Колесный диск', 'Катанка'],
        'ПК Челябинский МК': ['Болт М2', 'Штуцер М26хШ22', 'Калиброванный
прокат', 'Лента'],
        'ПК Металлогорский МК': ['Болт М2', 'Штуцер М26хШ22', 'Калиброванный
прокат', 'Лента'],
    }

    self.assertEqual(task3(self.providers, self.details, self.dets_provs),
key_fot_test)

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

```

Результат выполнения программы

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

f0max@Linux-VM:~/Code/BKIT_2021/RK2$ /bin/python3 /home/f0max/Code/BKIT_2021/RK2/tests.py
...
-----
Ran 3 tests in 0.005s

OK
f0max@Linux-VM:~/Code/BKIT_2021/RK2$ █

```