

МГТУ им. Н.Э. Баумана

Отчёт по РК №2  
по курсу «Парадигмы и Конструкции Языков Программирования»  
Вариант Д19

Руководитель  
Гапанюк Ю.Е.

Студент группы ИУ5-36Б  
Мохаммед М. Н.

2024 г.

**Задание:** 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.

2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

**Листинг:**

**(ref.py)**

```
from collections import defaultdict

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

class Part:
    def __init__(self, part_id, name, manufacturer_id):
        self.part_id = part_id
        self.name = name
        self.manufacturer_id = manufacturer_id

class ManufacturerEmployee:
    def __init__(self, employee_id, manufacturer_id):
        self.employee_id = employee_id
        self.manufacturer_id = manufacturer_id

class Employee:
    def __init__(self, employee_id, name):
        self.employee_id = employee_id
        self.name = name

def get_parts_and_manufacturers(parts, manufacturers):
    result = [
        (part.name, manufacturer.name)
        for part in parts
        for manufacturer in manufacturers
        if part.manufacturer_id == manufacturer.manufacturer_id and
        part.name.endswith("op")
    ]
    return result

def get_manufacturer_parts_count(parts):
    manufacturer_parts_count = defaultdict(int)
    for part in parts:
        manufacturer_parts_count[part.manufacturer_id] += 1
    return manufacturer_parts_count

def get_manufacturer_avg_parts(manufacturers, manufacturer_parts_count):
    manufacturer_avg_parts = [
        (manufacturer.name,
        manufacturer_parts_count[manufacturer.manufacturer_id])
        for manufacturer in manufacturers
    ]
    manufacturer_avg_parts_sorted = sorted(manufacturer_avg_parts, key=lambda
x: x[1], reverse=True)
    return manufacturer_avg_parts_sorted

def get_manufacturers_and_employees(manufacturers, employees,
manufacturer_employees):
    result = [
```

```

        (manufacturer.name, [employee.name for employee in employees if
any(me.employee_id == employee.employee_id and me.manufacturer_id ==
manufacturer.manufacturer_id for me in manufacturer_employees)])
        for manufacturer in manufacturers
        if manufacturer.name.startswith("К")
    ]
    return result

manufacturers = [
    Manufacturer(1, "Кировский завод"),
    Manufacturer(2, "Технопром"),
    Manufacturer(3, "КамАЗ"),
    Manufacturer(4, "Автопром")
]

parts = [
    Part(1, "Мотор", 1),
    Part(2, "Ротор", 2),
    Part(3, "Амортизатор", 3),
    Part(4, "Генератор", 1),
    Part(5, "Транзистор", 2)
]

employees = [
    Employee(1, "Алексей"),
    Employee(2, "Борис"),
    Employee(3, "Кирилл"),
    Employee(4, "Дмитрий"),
    Employee(5, "Евгений")
]

manufacturer_employees = [
    ManufacturerEmployee(1, 1),
    ManufacturerEmployee(2, 2),
    ManufacturerEmployee(3, 3),
    ManufacturerEmployee(4, 1),
    ManufacturerEmployee(5, 3)
]

result1 = get_parts_and_manufacturers(parts, manufacturers)
manufacturer_parts_count = get_manufacturer_parts_count(parts)
manufacturer_avg_parts_sorted = get_manufacturer_avg_parts(manufacturers,
manufacturer_parts_count)
result3 = get_manufacturers_and_employees(manufacturers, employees,
manufacturer_employees)

print("1) Детали, заканчивающиеся на 'ор', и их производители:")
for part_name, manufacturer_name in result1:
    print(f"Деталь: {part_name}, Производитель: {manufacturer_name}")

print("2) Производители и среднее количество производимых деталей:")
for manufacturer_name, avg_part_count in manufacturer_avg_parts_sorted:
    print(f"Производитель: {manufacturer_name}, Среднее количество деталей: {avg_part_count}")

print("3) Производители, начинающиеся с 'К', и их сотрудники:")
for manufacturer_name, employee_names in result3:
    print(f"Производитель: {manufacturer_name}, Сотрудники: {' '.join(employee_names)}")

```

## (tests.py)

```
import unittest
from ref import (Manufacturer, Part, get_parts_and_manufacturers,
get_manufacturers_and_employees, get_manufacturer_parts_count,
ManufacturerEmployee, defaultdict, Employee)
class TestMyModule(unittest.TestCase):
    def test_get_parts_and_manufacturers(self):
        manufacturers = [Manufacturer(1, "Производитель А"), Manufacturer(2,
"Производитель Б")]
        parts = [Part(1, "Деталь Аор", 1), Part(2, "Деталь Бор", 2), Part(3,
"Деталь С", 1)]
        expected = [('Деталь Аор', 'Производитель А'), ('Деталь Бор',
'Производитель Б')]
        self.assertEqual(get_parts_and_manufacturers(parts, manufacturers),
expected)

    def test_get_manufacturer_parts_count(self):
        parts = [Part(1, "Деталь 1", 1), Part(2, "Деталь 2", 1), Part(3,
"Деталь 3", 2)]
        expected = defaultdict(int, {1: 2, 2: 1})
        self.assertEqual(get_manufacturer_parts_count(parts), expected)

    def test_get_manufacturers_and_employees(self):
        manufacturers = [Manufacturer(1, "К завод"), Manufacturer(2,
"Технопром")]
        employees = [Employee(1, "Иван"), Employee(2, "Петр")]
        manufacturer_employees = [ManufacturerEmployee(1, 1),
ManufacturerEmployee(2, 2)]
        expected = [('К завод', ['Иван'])]
        self.assertEqual(get_manufacturers_and_employees(manufacturers,
employees, manufacturer_employees), expected)

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

## Анализ результатов:

```
Testing started at 23:00 ...
Launching unittests with arguments python -m unittest C:\Users\mokka\

1)Детали, заканчивающиеся на 'ор', и их производители:
Деталь: Мотор, Производитель: Кировский завод
Деталь: Ротор, Производитель: Технопром
Деталь: Амортизатор, Производитель: КамАЗ
Деталь: Генератор, Производитель: Кировский завод
Деталь: Транзистор, Производитель: Технопром
2)Производители и среднее количество производимых деталей:
Производитель: Кировский завод, Среднее количество деталей: 2
Производитель: Технопром, Среднее количество деталей: 2
Производитель: КамАЗ, Среднее количество деталей: 1
Производитель: Автопром, Среднее количество деталей: 0
3)Производители, начинающиеся с 'К', и их сотрудники:
Производитель: Кировский завод, Сотрудники: Алексей, Дмитрий
Производитель: КамАЗ, Сотрудники: Кирилл, Евгений

Ran 3 tests in 0.005s

OK

Process finished with exit code 0
```