



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

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

**Рубежный контроль № 2
по дисциплине «Базовые компоненты интернет-технологий»**

**Выполнила:
студентка группы ИУ5-33Б
Шаповалова В.В.**

**Проверил:
Гапанюк Ю.Е.**

2021 г.

Полученное 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).

Текст программы:

Файл main.py

```
from operator import itemgetter

class Op:
    """Оператор"""

    def __init__(self, id, op_name, memory, ln_id):
        self.id = id
        self.op_name = op_name
        self.memory = memory
        self.ln_id = ln_id

class Lng:
    """Язык программирования"""

    def __init__(self, id, name):
        self.id = id
        self.name = name

class OpLng:
    """
    Операторы языков программирования
    """

    def __init__(self, ln_id, op_id):
        self.ln_id = ln_id
        self.op_id = op_id

# Языки программирования
lngs = [
    Lng(1, 'Python'),
    Lng(2, 'C'),
    Lng(3, 'Java'),
    Lng(4, 'C++'),
]

# Операторы
ops = [
    Op(1, 'Сложение', 4, 1),
    Op(2, 'Умножение', 4, 2),
    Op(3, 'Сравнение', 2, 3),
    Op(4, 'Вызов функции', 8, 3),
]
```

```

# Языки и операторы, для связи многие-ко-многим
ln_op = [
    OpLng(1, 1),
    OpLng(2, 2),
    OpLng(3, 3),
    OpLng(3, 4),
    OpLng(1, 2),
    OpLng(1, 4),
    OpLng(2, 1),
    OpLng(2, 3),
    OpLng(3, 1),
    OpLng(3, 2),
    OpLng(1, 4),
    OpLng(4, 4),
]

def first_task(one_to_many):
    task1 = []
    for op_name, memory, name in one_to_many:
        if name[0] == "J":
            task1.append((name, op_name))
    return (task1)

def second_task(one_to_many):
    mas2 = []
    for ln in lngs:
        op_lns = list(filter(lambda i: i[2] == ln.name, one_to_many))
        if len(op_lns) > 0:
            ln_memory = [memory for _, memory, _ in op_lns]
            Memor = min(ln_memory)
            mas2.append((ln.name, Memor))
    task2 = sorted(mas2, key=itemgetter(1))
    return(task2)

def third_task(many_to_many):
    mas3 = []
    for op_name, memory, name in many_to_many:
        mas3.append((op_name, name))

    task3 = list(sorted(mas3, key=itemgetter(0)))
    return(task3)

def main():
    """Основная функция"""
    one_to_many = [(op.op_name, op.memory, ln.name)
                   for ln in lngs
                   for op in ops
                   if op.ln_id == ln.id]

    many_to_many_temp = [(ln.name, lop.ln_id, lop.op_id)
                         for ln in lngs
                         for lop in ln_op
                         if ln.id == lop.ln_id]
    many_to_many = [(op.op_name, op.memory, language)
                    for language, ln_id, op_id in many_to_many_temp
                    for op in ops if op.id == op_id]

    print('\nЗадание 1 \n', first_task(one_to_many))
    print('\nЗадание 2 \n', second_task(one_to_many))
    print('\nЗадание 3 \n', third_task(many_to_many))

```

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

Файл test.py

```
import unittest
import sys, os
sys.path.append(os.getcwd())
from main import *

class TddTest(unittest.TestCase):
    def test_first_task(self):
        one_to_many = [(op.op_name, op.memory, ln.name)
                        for ln in lngs
                        for op in ops
                        if op.ln_id == ln.id]
        self.assertEqual(first_task(one_to_many), [('Java', 'Сравнение'),
        ('Java', 'Вызов функции')])
    def test_2(self):
        one_to_many = [(op.op_name, op.memory, ln.name)
                        for ln in lngs
                        for op in ops
                        if op.ln_id == ln.id]
        self.assertEqual(second_task(one_to_many), [('Java', 2), ('Python',
4), ('C', 4)])
    def test_3(self):
        many_to_many_temp = [(ln.name, lop.ln_id, lop.op_id)
                              for ln in lngs
                              for lop in ln.op
                              if ln.id == lop.ln_id]
        many_to_many = [(op.op_name, op.memory, language)
                          for language, ln_id, op_id in many_to_many_temp
                          for op in ops if op.id == op_id]
        self.assertEqual(third_task(many_to_many), [('Вызов функции', 'Python'),
        ('Вызов функции', 'Python'), ('Вызов функции', 'Java'), ('Вызов функции',
        'C++'), ('Сложение', 'Python'), ('Сложение', 'C'), ('Сложение', 'Java'),
        ('Сравнение', 'C'), ('Сравнение', 'Java'), ('Умножение', 'Python'),
        ('Умножение', 'C'), ('Умножение', 'Java')])
```

Результат выполнения:

```
Testing started at 1:56 ...
```

```
Ran 3 tests in 0.002s
```

```
OK
```

```
Launching unittests with arguments python -m unittest C:/Users/Хезер Чендлер/PycharmProjects/pythonProj
```

```
Process finished with exit code 0
```