



**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»**

**Выполнил: студент группы ИУ5-35Б
Юриков Е.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).

Листинг программы

```
import unittest

class Language:
    def __init__(self, lang_id, name):
        self.lang_id = lang_id
        self.name = name

# Языки
languages = [Language(1, 'C++'),
              Language(2, 'Python'),
              Language(3, 'Java'),
              Language(4, 'C#')]

class Operator:
    def __init__(self, op_id, name, uses, lang_id):
        self.op_id = op_id
        self.name = name
        self.uses = uses
        self.lang_id = lang_id

# Названия
operators = [Operator(1, 'сложение', 95, 1),
              Operator(2, 'вычитание', 95, 1),
              Operator(3, 'противоположное число', 42, 4),
              Operator(4, 'возведение в степень', 20, 2),
              Operator(5, 'присваивание', 100, 3)]

class OpLang:
    def __init__(self, lang_id, op_id):
        self.lang_id = lang_id
        self.op_id = op_id

op_lang = [OpLang(1, 1),
            OpLang(1, 2),
            OpLang(4, 3),
            OpLang(2, 4),
            OpLang(3, 5)]

def find(element, arr):
    for i in arr:
        if i.name == element:
            return i.lang_id

one_to_many = [(op.name, op.uses, lan.name)
                for op in operators
                 for lan in languages
                 if op.lang_id == lan.lang_id]
```

```

        for lan in languages
        if op.lang_id == lan.lang_id]

many_to_many_tmp = [(lan.name, ol.lang_id, ol.op_id)
                    for lan in languages
                    for ol in op_lang
                    if lan.lang_id == ol.lang_id]

many_to_many = [(lan_name, op.name)
                for lan_name, lang_id, op_id in many_to_many_tmp
                for op in operators
                if op.op_id == op_id]

# Д1

def t1(one_to_many_f):
    print('Задание 1')
    ans1 = []
    for el in one_to_many_f:
        if el[0][len(el[0]) - 2:] == 'ие':
            ans1.append((el[0], el[2]))
    return ans1

def t2(languages_f, operators_f):
    print('Задание 2')
    sum = [0 for i in range(0, len(languages_f))]
    count = [0 for i in range(0, len(languages_f))]
    for i in operators_f:
        sum[i.lang_id - 1] += i.uses
        count[i.lang_id - 1] += 1

    ans2 = []
    for i in range(len(sum)):
        ans2.append((languages[i].name, sum[i] / count[i]))

    return sorted(ans2, key = lambda s: s[1])

def t3(languages_f, many_to_many_f):
    print('Задание 3')
    ans3 = {}
    _languages = list(filter(lambda x: x.name[0] == 'C', languages_f))

    for i in _languages:
        anstmp = []
        tmp = list(filter(lambda x: x[0] == i.name, many_to_many_f))
        for j in tmp:
            anstmp.append(j[1])
        ans3[i.name] = anstmp
    return ans3

class Test(unittest.TestCase):

    def test_task1(self):

        test_op = Operator(1, 'сложение', 95, 1)
        test_lang = Language(1, 'C++')
        test_one_to_many = [(test_op.name, test_op.uses, test_lang.name)]

        ans = t1(test_one_to_many)

```

```

        self.assertTrue(ans, [('сложение', 'C++')])

    def test_task2(self):

        test_op = [Operator(1, 'сложение', 95, 1)]
        test_lang = [Language(1, 'C++')]

        ans = t2(test_lang, test_op)

        self.assertTrue(ans, [('C++', 95)])

    def test_task3(self):

        test_op = [Operator(1, 'сложение', 95, 1)]
        test_lang = [Language(1, 'C++')]
        test_many_to_many = [(test_lang[0].name, test_op[0].name)]
        ans = t3(test_lang, test_many_to_many)

        self.assertTrue(ans, {'C++': ['сложение']})

def main():

    # Д1

    print(t1(one_to_many))

    # Д2

    print(t2(languages, operators))

    # Д3

    print(t3(languages, many_to_many))

if __name__ == '__main__':
    main()

```

Вывод

```

C:\Users\egory\Desktop\RK2\venv\Scripts\python.exe "D:\PyCharm Community Edition 2021.2.2\plugins\python-ce\helpers\pycharm\_jb_unittest_runner
Testing started at 21:51 ...
Launching unittests with arguments python -m unittest C:/Users/egory/Desktop/RK2/main.py in C:\Users\egory\Desktop\RK2

Ran 3 tests in 0.015s

OK

Process finished with exit code 0
Задание 1
Задание 2
Задание 3

```