

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

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

**Курс «Парадигмы и конструкции языков программирования»**

**Отчет по Рубежному контролю 2  
«Нахождение корней нелинейного уравнения»  
Вариант №21 Б**

Выполнил:  
студент группы ИУ5-33Б:  
Бегдаш С. Ю.  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю.Е.  
Подпись и дата:

Москва, 2025 г.

```

main.py:
import unittest
from functions import *
from data import *

class TestOperatorFunctions(unittest.TestCase):
    def setUp(self):
        self.languages = langs
        self.operators = ops
        self.ops_langs = ops_langs
        self.one_to_many = get_one_to_many(langs, ops)
        self.many_to_many = get_many_to_many(ops, langs, ops_langs)

    def test_func1(self):
        res_b1 = func1(self.one_to_many)
        self.assertEqual(res_b1[0], ('breakov', 'C++'))

    def test_func2(self):
        res_b2 = func2(self.one_to_many, self.languages)
        self.assertIn(('Python', 2), res_b2)
        self.assertIn(('C++', 2), res_b2)
        self.assertIn(('Java', 1), res_b2)

    def test_func3(self):
        res_b3 = func3(self.many_to_many, "ov")
        self.assertTrue(('breakov', 'C++') in res_b3)
        self.assertTrue(('breakov', 'Java') in res_b3)

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

```

functions.py:

```

def get_one_to_many(languages, operators):
    return [(op.name, lang.title)
            for lang in languages
            for op in operators
            if op.lang_id == lang.id]

def get_many_to_many(operators, languages, ops_langs):
    temp = [(ol.op_id, ol.lang_id) for ol in ops_langs]
    return [(op.name, lang.title)
            for op_id, lang_id in temp
            for op in operators if op.id == op_id
            for lang in languages if lang.id == lang_id]

def func1(one_to_many):
    return sorted(one_to_many, key=lambda x: x[0])

```

```

def func2(one_to_many, languages):
    res = []
    for lang in languages:
        ops_count = len([x for x in one_to_many if x[1] == lang.title])
        if ops_count > 0:
            res.append((lang.title, ops_count))
    return sorted(res, key=lambda x: x[1], reverse=True)

def func3(many_to_many, suffix):
    return [(op_name, lang_title) for op_name, lang_title in many_to_many if
    op_name.endswith(suffix)]

data.py:
from classes import *

langs = [
    Language(1, "Python"),
    Language(2, "C++"),
    Language(3, "Java"),
]

ops = [
    Operator(1, "print", 1),
    Operator(2, "cout", 2),
    Operator(3, "for", 1),
    Operator(4, "if", 3),
    Operator(5, "breakov", 2),
]

ops_langs = [
    OperatorLanguage(1, 1),
    OperatorLanguage(2, 2),
    OperatorLanguage(3, 1),
    OperatorLanguage(4, 3),
    OperatorLanguage(5, 2),
    OperatorLanguage(5, 3),
]

classes.py:
class Operator:
    def __init__(self, id, name, lang_id):
        self.id = id
        self.name = name
        self.lang_id = lang_id

class Language:
    def __init__(self, id, title):
        self.id = id
        self.title = title

class OperatorLanguage:
    def __init__(self, op_id, lang_id):

```

```
self.op_id = op_id  
self.lang_id = lang_id
```

Вывод:

```
s/pythonProjects/rk2/main.py
```

```
...
```

```
-----
```

```
Ran 3 tests in 0.001s
```

```
OK
```