

**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 1
«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:
from functions import func1, func2, func3
from data import *

def main():
    # Связь один-ко-многим
    one_to_many = [(op.name, lang.title)
                    for lang in langs
                    for op in ops
                    if op.lang_id == lang.id]

    # Связь многие-ко-многим
    many_to_many_temp = [(ol.op_id, ol.lang_id) for ol in ops_langs]
    many_to_many = [(op.name, lang.title)
                     for op_id, lang_id in many_to_many_temp
                     for op in ops if op.id == op_id
                     for lang in langs if lang.id == lang_id]
    print(f'Б1: {func1(one_to_many)}\nБ2: {func2(one_to_many, langs)}\nБ3: {func3(many_to_many)}')

if __name__ == '__main__':
    main()

```

```

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

```

```

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

```

```

# Для связи многие-ко-многим
class OperatorLanguage:
    def __init__(self, op_id, lang_id):
        self.op_id = op_id
        self.lang_id = lang_id

```

```

functions.py:
# Функция для Б1
def func1(one_to_many : list) -> list:
    return sorted(one_to_many, key=lambda x: x[1])

```

```

# Функция для Б2
def func2(one_to_many : list, langs : list) -> list:
    res_b2_unsorted = []

```

```

for lang in langs:
    ops_count = len([x for x in one_to_many if x[1] == lang.title])
    if ops_count > 0:
        res_b2_unsorted.append((lang.title, ops_count))
res_b2 = sorted(res_b2_unsorted, key=lambda x: x[1], reverse=True)
return res_b2

# Функция для БЗ
def func3(many_to_many : list) -> list:
    return [(op_name, lang_title) for op_name, lang_title in many_to_many if
op_name.endswith("ov")]

data.py:
from classes import *

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

# Операторы
ops = [
    Operator(1, "print", 1),
    Operator(2, "cout", 2),
    Operator(3, "for", 1),
    Operator(4, "if", 3),
    Operator(5, "breakov", 2), # пример фамилии/оператора на "ов"
]

# Связи многие-ко-многим, у одного оператора может быть много языков
программирования
ops_langs = [
    OperatorLanguage(1, 1),
    OperatorLanguage(2, 2),
    OperatorLanguage(3, 1),
    OperatorLanguage(4, 3),
    OperatorLanguage(5, 2),
    OperatorLanguage(5, 3),
]

Выводы программы:
Б1: [('cout', 'C++'), ('breakov', 'C++'), ('if', 'Java'), ('print', 'Python'), ('for', 'Python')]
Б2: [('Python', 2), ('C++', 2), ('Java', 1)]
Б3: [('breakov', 'C++'), ('breakov', 'Java')]

```

