

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н.Э.  
Баумана

Факультет “Информатика и системы управления”  
Кафедра “Системы обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по рубежному контролю №1

**Выполнил:**

Студент группы ИУ5-34Б

Изотов Е.А

**Преподаватель:**

Гапанюк Ю.Е.

Москва 2025

```
class ProgrammingLanguage:
```

```
    def __init__(self, id_lang, name):
```

```
        self.id_lang = id_lang
```

```
        self.name = name
```

```
class Operator:
```

```
    def __init__(self, id_op, content, length, lang_id):
```

```
        self.id_op = id_op
```

```
        self.content = content
```

```
        self.length = length
```

```
        self.lang_id = lang_id
```

```
class OperatorLanguage:
```

```
    def __init__(self, id_op_lang, operator_id, language_id):
```

```
        self.id_op_lang = id_op_lang
```

```
        self.operator_id = operator_id
```

```
        self.language_id = language_id
```

```
languages = [
```

```
    ProgrammingLanguage(1, "Go"),
```

```
    ProgrammingLanguage(2, "Rust"),
```

```
    ProgrammingLanguage(3, "Java")
```

```
]
```

```

operators = [
    Operator(1, "break", 5, 1),
    Operator(2, "borrow", 6, 2),
    Operator(3, "object", 6, 3),
    Operator(4, "guard", 5, 2)
]

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

# Вывести все операторы, которые начинаются с "b", и их языки
# программирования

def RequestOne(operators, languages):
    return [
        (op.content, next(l.name for l in languages if l.id_lang == op.lang_id))
        for op in operators if op.content.startswith("b")
    ]

# Получить список длин операторов (кол-во символов),
# отсортированный по возрастанию

def RequestTwo(operators):
    # Используем map + sorted
    return sorted(map(lambda op: op.length, operators))

```

```

# Вывести все операторы и их языки (многие-ко-многим)

def RequestThree(operators, languages, operators_languages):

    # Используем list comprehension + сортировку

    result = [

        (op.id_op, op.content, lang.name)

        for ol in operators_languages

        for op in operators if op.id_op == ol.operator_id

        for lang in languages if lang.id_lang == ol.language_id

    ]

    return sorted(result, key=lambda x: x[0])


print(" Операторы, начинающиеся с 'b', и их языки программирования:")

for content, lang in RequestOne(operators, languages):

    print(f"Оператор: {content}, Язык: {lang}")


print("\n Список длин операторов (по возрастанию):")

print(RequestTwo(operators))


print("\n Все операторы и их языки (многие-ко-многим):")

for op_id, content, lang in RequestThree(operators, languages,
operators_languages):

    print(f"ID: {op_id}, Оператор: {content}, Язык: {lang}")

```

## Работа кода:

```
Операторы, начинающиеся с 'b', и их языки программирования:  
Оператор: break, Язык: Go  
Оператор: borrow, Язык: Rust
```

```
Список длин операторов (по возрастанию):  
[5, 5, 6, 6]
```

```
Все операторы и их языки (многие-ко-многим):  
ID: 1, Оператор: break, Язык: Go  
ID: 2, Оператор: borrow, Язык: Rust  
ID: 3, Оператор: object, Язык: Java  
ID: 3, Оператор: object, Язык: Rust  
ID: 4, Оператор: guard, Язык: Rust
```

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