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

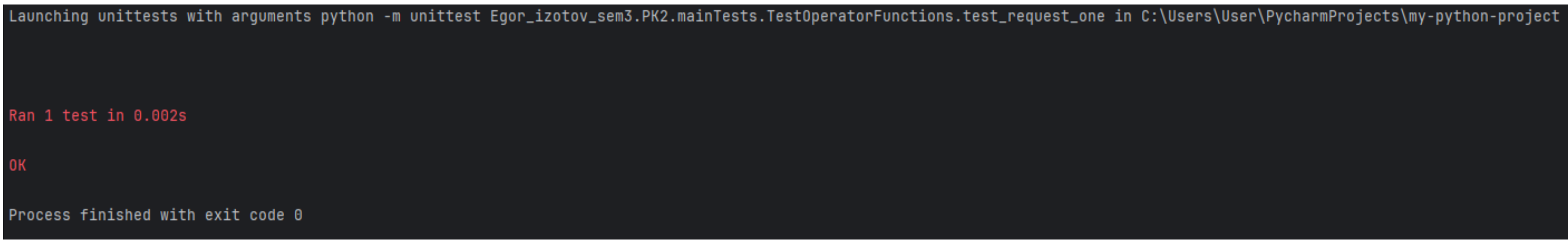
Код(main.py)

```
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


# Функция 1: 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")
    ]


# Функция 2: Получить список длин операторов (кол-во символов), отсортированный по возрастанию
def RequestTwo(operators):
    return sorted(map(lambda op: op.length, operators))


# Функция 3: Вывести все операторы и их языки (многие-ко-многим)
def RequestThree(operators, languages, operators_languages):
    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])


# Для удобства тестирования создадим функцию для получения тестовых данных
def get_sample_data():
    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)
    ]

    return languages, operators, operators_languages
```

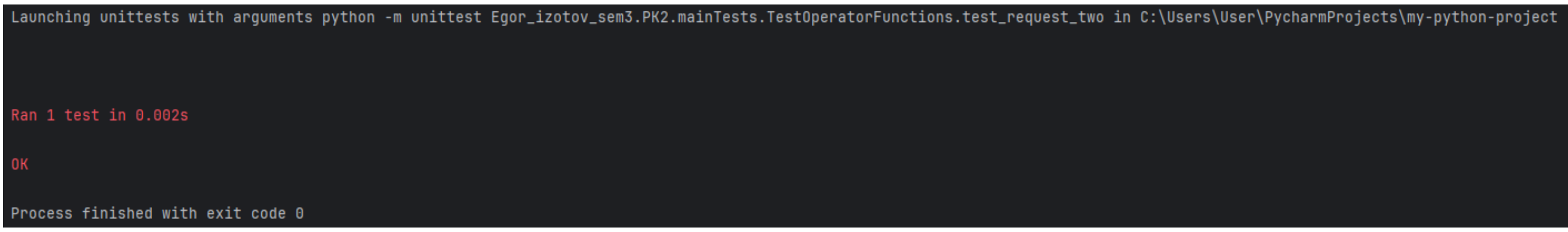
Код(mainTesta.py)

```
import unittest

from main import RequestOne, RequestTwo, RequestThree, get_sample_data,
ProgrammingLanguage, \
    Operator, OperatorLanguage


class TestOperatorFunctions(unittest.TestCase):
    def setUp(self):
        # Здесь создаем все данные, как раньше
        self.languages = [
            ProgrammingLanguage(1, "Go"),
            ProgrammingLanguage(2, "Rust"),
            ProgrammingLanguage(3, "Java")
        ]
        self.operators = [
            Operator(1, "break", 5, 1),
            Operator(2, "borrow", 6, 2),
            Operator(3, "object", 6, 3),
            Operator(4, "guard", 5, 2)
        ]
        self.operators_languages = [
            OperatorLanguage(1, 1, 1),
            OperatorLanguage(2, 2, 2),
            OperatorLanguage(3, 3, 3),
            OperatorLanguage(4, 4, 2),
            OperatorLanguage(5, 3, 2)
        ]

    def test_request_one(self):
        result = RequestOne(self.operators, self.languages)
        expected = [("break", "Go"), ("borrow", "Rust")]
        self.assertEqual(result, expected)

    def test_request_two(self):
        result = RequestTwo(self.operators)
        expected = [5, 5, 6, 6]
        self.assertEqual(result, expected)

    def test_request_three(self):
        result = RequestThree(self.operators, self.languages, self.operators_languages)
        expected = [
            (1, "break", "Go"),
            (2, "borrow", "Rust"),
            (3, "object", "Java"),
            (3, "object", "Rust"),
            (4, "guard", "Rust")
        ]
        self.assertEqual(result, expected)


if __name__ == "__main__":
    unittest.main()
```

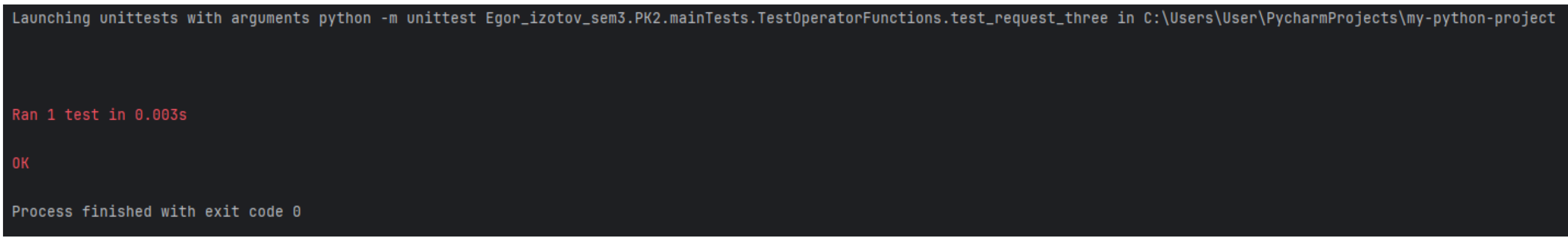
Работа кода:



```
Launching unittests with arguments python -m unittest Egor_izotov_sem3.PK2.mainTests.TestOperatorFunctions.test_request_one in C:\Users\User\PycharmProjects\my-python-project



Ran 1 test in 0.002s

OK

Process finished with exit code 0
```



```
Launching unittests with arguments python -m unittest Egor_izotov_sem3.PK2.mainTests.TestOperatorFunctions.test_request_two in C:\Users\User\PycharmProjects\my-python-project



Ran 1 test in 0.002s

OK

Process finished with exit code 0
```



```
Launching unittests with arguments python -m unittest Egor_izotov_sem3.PK2.mainTests.TestOperatorFunctions.test_request_three in C:\Users\User\PycharmProjects\my-python-project



Ran 1 test in 0.003s

OK

Process finished with exit code 0
```