

## Рефактор текста программы РК1:

```
class Program:
    def __init__(self, program_id, name, version, computer_id):
        self.program_id = program_id
        self.name = name
        self.version = version
        self.computer_id = computer_id

class Computer:
    def __init__(self, computer_id, name):
        self.computer_id = computer_id
        self.name = name

class ProgramsOnComputer:
    def __init__(self, program_id, computer_id):
        self.program_id = program_id
        self.computer_id = computer_id

def get_programs_ending_with_r(programs, computers):
    return [
        (p.name, c.name)
        for p in programs
        for c in computers
        if p.computer_id == c.computer_id and p.name.endswith('r')
    ]

def calculate_average_version(programs, computers):
    result = []
    for c in computers:
        comp_programs = [p for p in programs if p.computer_id == c.computer_id]
        if comp_programs:
            avg_version = sum(
                [float(p.version.split('v')[1]) for p in comp_programs]) / len(
                    comp_programs)
            result.append((c.name, avg_version))
    result.sort(key=lambda x: x[1])
    return result

def get_computers_with_programs_starting_with_a(computers, programs):
    return [
        (c.name, [p.name for p in programs if p.computer_id == c.computer_id])
        for c in computers if c.name.startswith('A')
    ]
```

## Модульные Тесты:

```
import unittest

from main import get_computers_with_programs_starting_with_a, \
    calculate_average_version, get_programs_ending_with_r, Program, Computer

class TestProgramAnalysis(unittest.TestCase):
    def setUp(self):
        self.computers = [
            Computer(1, 'Alpha Computer'),
            Computer(2, 'Beta Computer'),
            Computer(3, 'Aqua Computer')
        ]

        self.programs = [
            Program(1, 'Text Editor', 'v1.0', 1),
            Program(2, 'Browser', 'v2.1', 2),
            Program(3, 'Media Player', 'v3.5', 3),
            Program(4, 'Image Editor', 'v1.2', 1),
            Program(5, 'Video Editor', 'v4.0', 2)
        ]

    def test_get_programs_ending_with_r(self):
        result = get_programs_ending_with_r(self.programs, self.computers)
        expected = [('Text Editor', 'Alpha Computer'),
                    ('Browser', 'Beta Computer'),
                    ('Media Player', 'Aqua Computer'),
                    ('Image Editor', 'Alpha Computer'),
                    ('Video Editor', 'Beta Computer')]
        self.assertEqual(result, expected)

    def test_calculate_average_version(self):
        result = calculate_average_version(self.programs, self.computers)
        expected = [('Alpha Computer', 1.1), ('Beta Computer', 3.05),
                    ('Aqua Computer', 3.5)]
        self.assertEqual(result, expected)

    def test_get_computers_with_programs_starting_with_a(self):
        result = get_computers_with_programs_starting_with_a(self.computers,
                                                                self.programs)

        expected = [
            ('Alpha Computer', ['Text Editor', 'Image Editor']),
            ('Aqua Computer', ['Media Player'])
        ]
        self.assertEqual(result, expected)

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