

Текст программы:

```
import unittest
class Computer:
    def __init__(self, id, name, department):
        self.id = id
        self.name = name
        self.department = department
        self.programs = []

    def add_program(self, program):
        if program not in self.programs:
            self.programs.append(program)
            program.add_computer(self)

class Program:
    def __init__(self, id, name, performance):
        self.id = id
        self.name = name
        self.performance = performance
        self.computers = []

    def add_computer(self, computer):
        if computer not in self.computers:
            self.computers.append(computer)

def get_computers_sorted_by_department(computers):
    return sorted(computers, key=lambda c: c.department)

def get_computers_sorted_by_total_performance(computers):
    return sorted(computers, key=lambda c: sum(p.performance for p in c.programs), reverse=True)

def get_computers_by_name_substring(computers, substring):
    return [computer for computer in computers if substring in computer.name]

class TestComputerProgramRelations(unittest.TestCase):
    def setUp(self):
        self.computer1 = Computer(1, "Компьютер Alpha", "Отдел разработки")
        self.computer2 = Computer(2, "Компьютер Beta", "Отдел аналитики")
        self.computer3 = Computer(3, "Компьютер Gamma", "Отдел исследований")

        self.program1 = Program(1, "Программа A", 50)
        self.program2 = Program(2, "Программа B", 75)
        self.program3 = Program(3, "Программа C", 100)
        self.program4 = Program(4, "Программа D", 60)

        self.computer1.add_program(self.program1)
        self.computer1.add_program(self.program2)
        self.computer2.add_program(self.program3)
        self.computer3.add_program(self.program4)

        self.computers = [self.computer1, self.computer2, self.computer3]
```

```

def test_sort_by_department(self):
    result = get_computers_sorted_by_department(self.computers)
    self.assertEqual([c.department for c in result], ["Отдел аналитики", "Отдел исследований",
"Отдел разработки"])

def test_sort_by_total_performance(self):
    result = get_computers_sorted_by_total_performance(self.computers)
    self.assertEqual([c.name for c in result], ["Компьютер Alpha", "Компьютер Beta", "Компьютер
Gamma"])

def test_filter_by_name_substring(self):
    result = get_computers_by_name_substring(self.computers, "Компьютер")
    self.assertEqual([c.name for c in result], ["Компьютер Alpha", "Компьютер Beta", "Компьютер
Gamma"])

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

```

