Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «ИНЖЕНЕРНЫЙ БИЗНЕС И МЕНЕДЖМЕНТ»

КАФЕДРА «ПРОМЫШЛЕННАЯ ЛОГИСТИКА» (ИБМ-3)

ОТЧЕТ По рк №»2

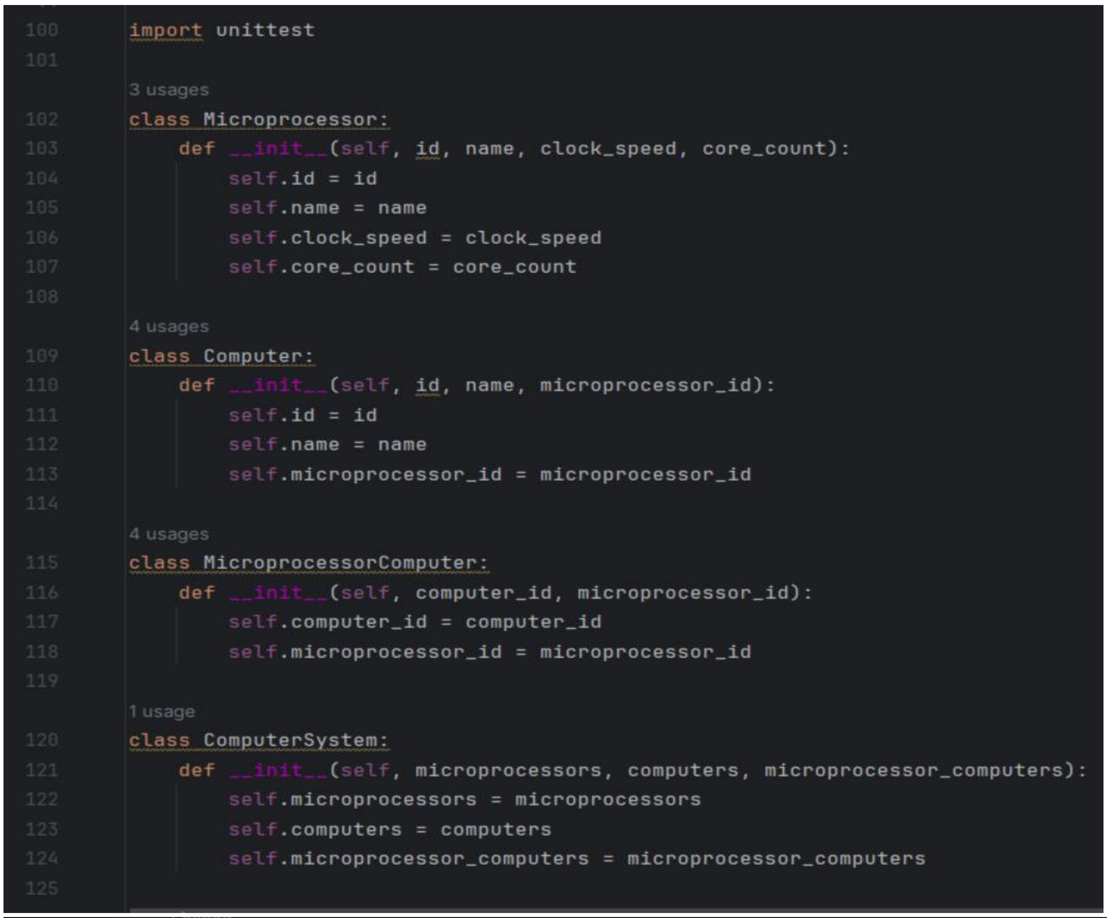
По дисциплине «Парадигмы и конструкции языков программирования»

«Вариант 7»

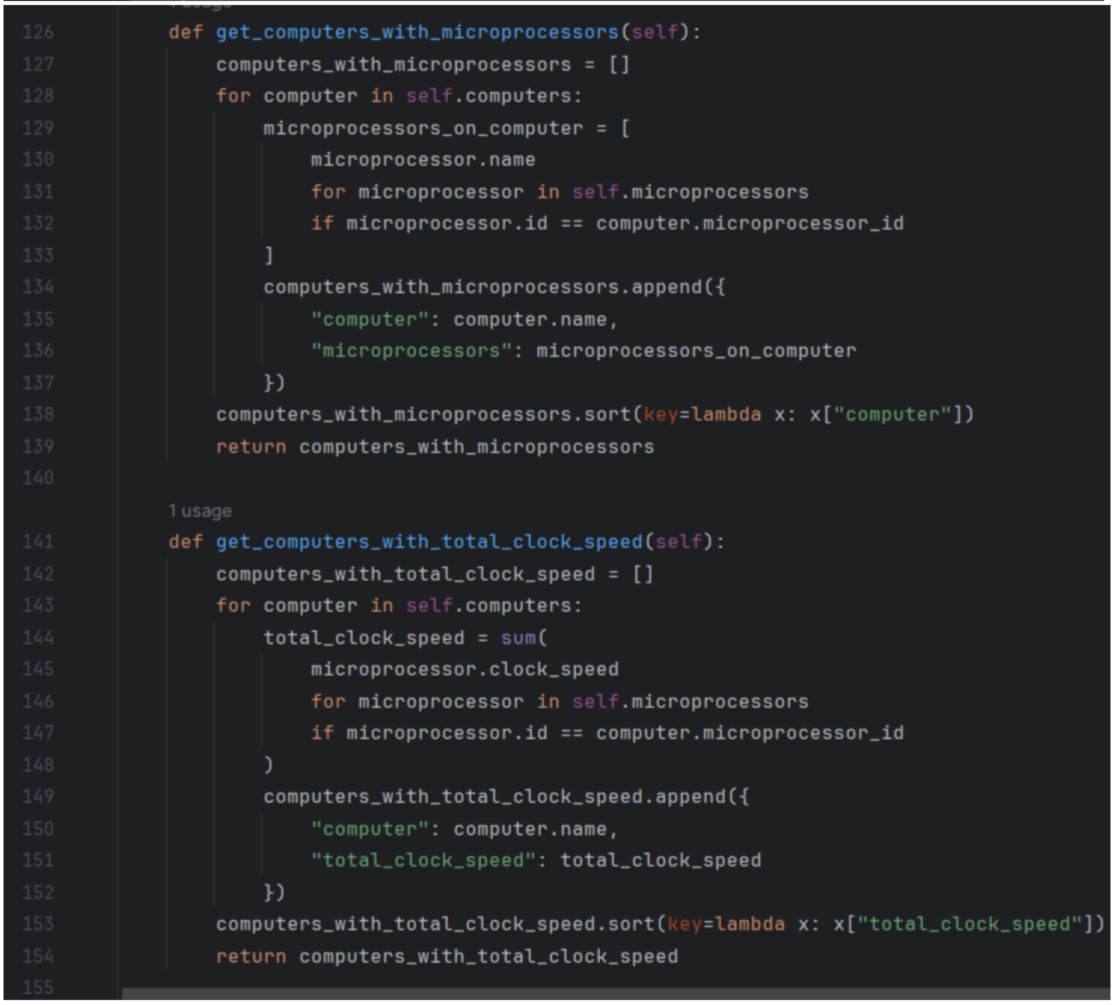
Студент ИБМ3-34Б Булюк М.Д.

Руководитель Гапанюк Ю.Е.

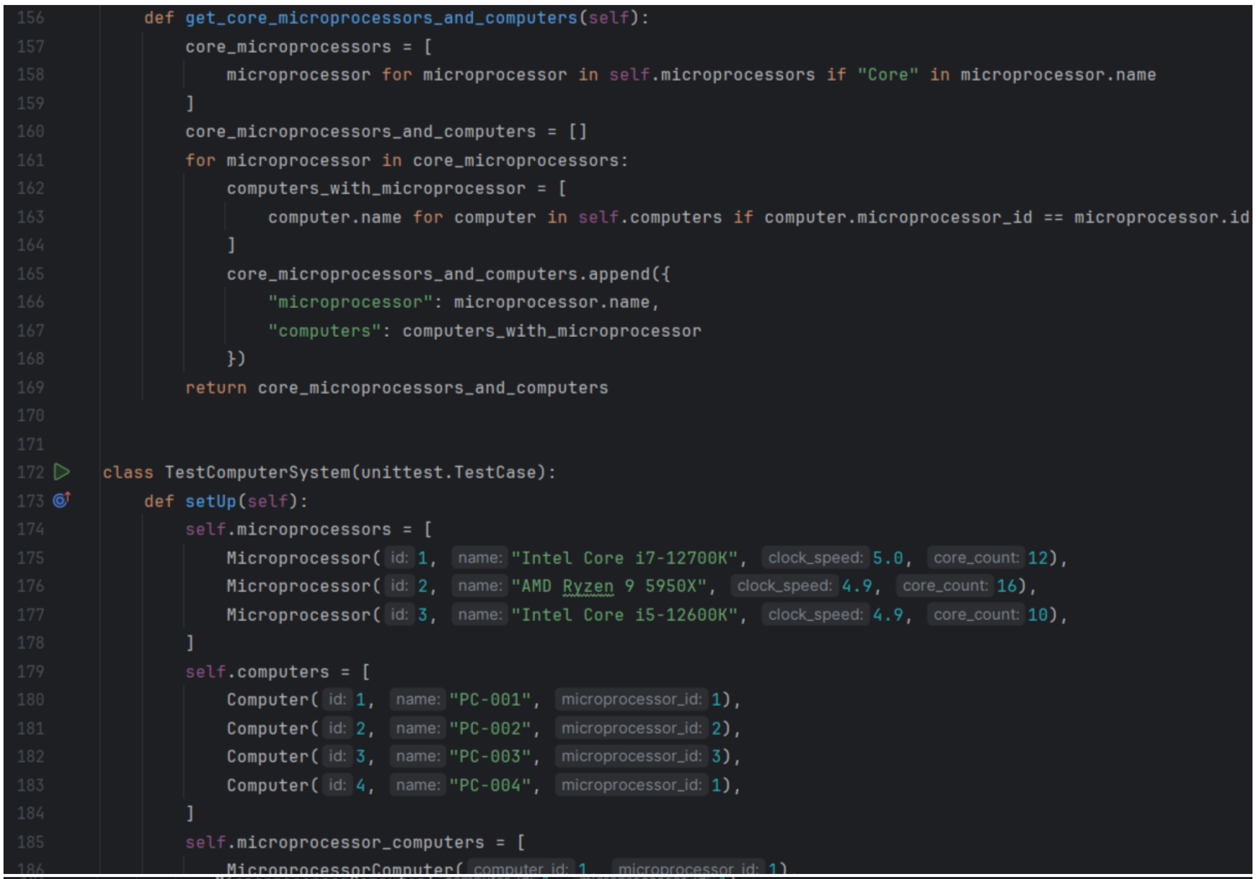
2024 г.



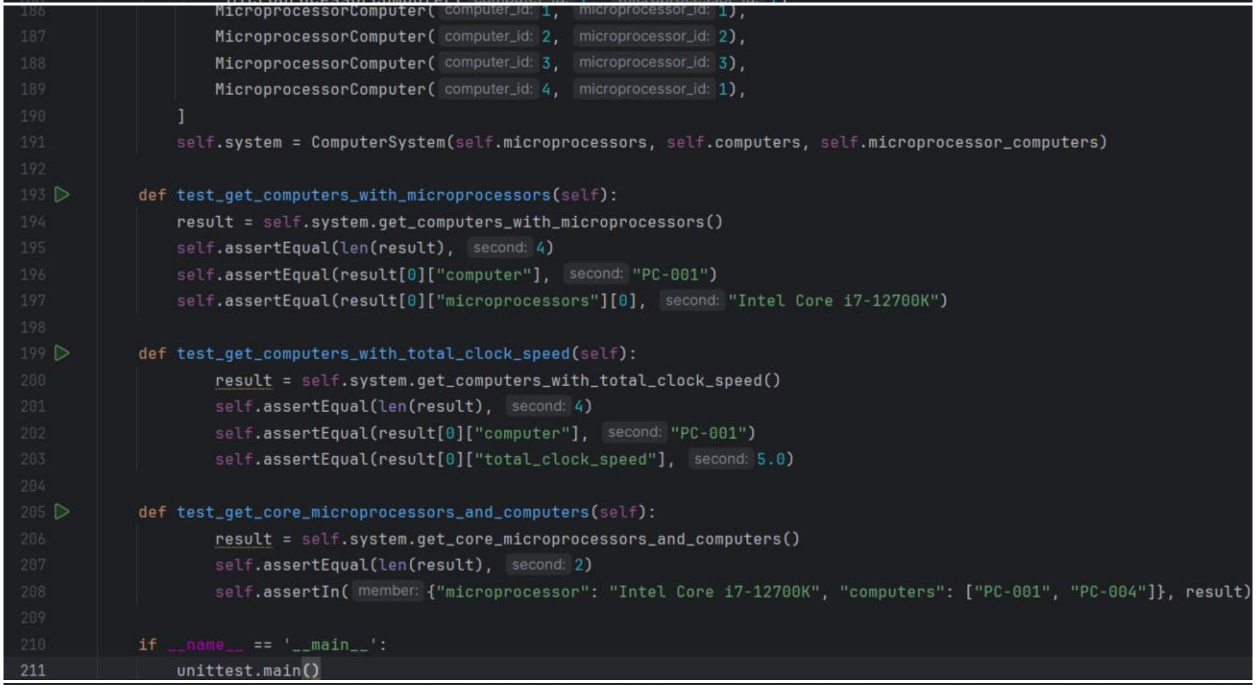
```python
import unittest

3 usages
class Microprocessor:
    def __init__(self, id, name, clock_speed, core_count):
        self.id = id
        self.name = name
        self.clock_speed = clock_speed
        self.core_count = core_count

4 usages
class Computer:
    def __init__(self, id, name, microprocessor_id):
        self.id = id
        self.name = name
        self.microprocessor_id = microprocessor_id

4 usages
class MicroprocessorComputer:
    def __init__(self, computer_id, microprocessor_id):
        self.computer_id = computer_id
        self.microprocessor_id = microprocessor_id

1 usage
class ComputerSystem:
    def __init__(self, microprocessors, computers, microprocessor_computers):
        self.microprocessors = microprocessors
        self.computers = computers
        self.microprocessor_computers = microprocessor_computers

```



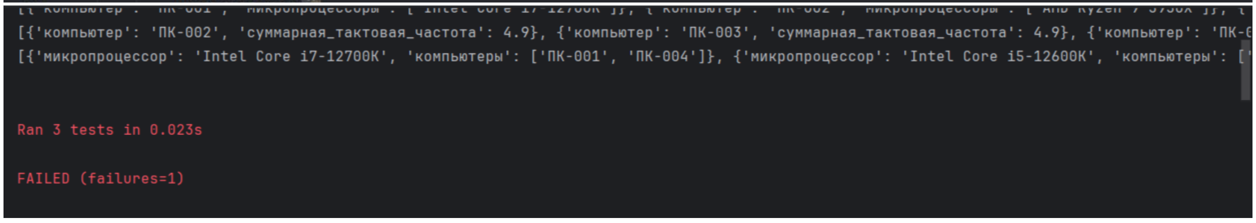
```python
    def get_computers_with_microprocessors(self):
        computers_with_microprocessors = []
        for computer in self.computers:
            microprocessors_on_computer = [
                microprocessor.name
                for microprocessor in self.microprocessors
                if microprocessor.id == computer.microprocessor_id
            ]
            computers_with_microprocessors.append({
                "computer": computer.name,
                "microprocessors": microprocessors_on_computer
            })
        computers_with_microprocessors.sort(key=lambda x: x["computer"])
        return computers_with_microprocessors

    1 usage
    def get_computers_with_total_clock_speed(self):
        computers_with_total_clock_speed = []
        for computer in self.computers:
            total_clock_speed = sum(
                microprocessor.clock_speed
                for microprocessor in self.microprocessors
                if microprocessor.id == computer.microprocessor_id
            )
            computers_with_total_clock_speed.append({
                "computer": computer.name,
                "total_clock_speed": total_clock_speed
            })
        computers_with_total_clock_speed.sort(key=lambda x: x["total_clock_speed"])
        return computers_with_total_clock_speed

```



```python
    def get_core_microprocessors_and_computers(self):
        core_microprocessors = [
            microprocessor for microprocessor in self.microprocessors if "Core" in microprocessor.name
        ]
        core_microprocessors_and_computers = []
        for microprocessor in core_microprocessors:
            computers_with_microprocessor = [
                computer.name for computer in self.computers if computer.microprocessor_id == microprocessor.id
            ]
            core_microprocessors_and_computers.append({
                "microprocessor": microprocessor.name,
                "computers": computers_with_microprocessor
            })
        return core_microprocessors_and_computers


class TestComputerSystem(unittest.TestCase):
    def setUp(self):
        self.microprocessors = [
            Microprocessor( id: 1,  name: "Intel Core i7-12700K",  clock_speed: 5.0,  core_count: 12),
            Microprocessor( id: 2,  name: "AMD Ryzen 9 5950X",  clock_speed: 4.9,  core_count: 16),
            Microprocessor( id: 3,  name: "Intel Core i5-12600K",  clock_speed: 4.9,  core_count: 10),
        ]
        self.computers = [
            Computer( id: 1,  name: "PC-001",  microprocessor_id: 1),
            Computer( id: 2,  name: "PC-002",  microprocessor_id: 2),
            Computer( id: 3,  name: "PC-003",  microprocessor_id: 3),
            Computer( id: 4,  name: "PC-004",  microprocessor_id: 1),
        ]
        self.microprocessor_computers = [
```



```python
            MicroprocessorComputer( computer_id: 1,  microprocessor_id: 1),
            MicroprocessorComputer( computer_id: 2,  microprocessor_id: 2),
            MicroprocessorComputer( computer_id: 3,  microprocessor_id: 3),
            MicroprocessorComputer( computer_id: 4,  microprocessor_id: 1),
        ]
        self.system = ComputerSystem(self.microprocessors, self.computers, self.microprocessor_computers)

    def test_get_computers_with_microprocessors(self):
        result = self.system.get_computers_with_microprocessors()
        self.assertEqual(len(result),  second: 4)
        self.assertEqual(result[0]["computer"],  second: "PC-001")
        self.assertEqual(result[0]["microprocessors"][0],  second: "Intel Core i7-12700K")

    def test_get_computers_with_total_clock_speed(self):
            result = self.system.get_computers_with_total_clock_speed()
            self.assertEqual(len(result),  second: 4)
            self.assertEqual(result[0]["computer"],  second: "PC-001")
            self.assertEqual(result[0]["total_clock_speed"],  second: 5.0)

    def test_get_core_microprocessors_and_computers(self):
            result = self.system.get_core_microprocessors_and_computers()
            self.assertEqual(len(result),  second: 2)
            self.assertIn( member: {"microprocessor": "Intel Core i7-12700K", "computers": ["PC-001", "PC-004"]}, result)

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



```
[{'компьютер': 'ПК-001', 'микропроцессоры': ['Intel Core i7-12700K']}, {'компьютер': 'ПК-002', 'микропроцессоры': ['AMD Ryzen 9 5950X']}, {
[{'компьютер': 'ПК-002', 'суммарная_тактовая_частота': 4.9}, {'компьютер': 'ПК-003', 'суммарная_тактовая_частота': 4.9}, {'компьютер': 'ПК-0
[{'микропроцессор': 'Intel Core i7-12700K', 'компьютеры': ['ПК-001', 'ПК-004']}, {'микропроцессор': 'Intel Core i5-12600K', 'компьютеры': ['


Ran 3 tests in 0.023s

FAILED (failures=1)
```