

**Московский государственный технический университет
им. Н.Э. Баумана**

**Факультет «Радиотехнический»
Кафедра «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

Рубежный контроль №2

Вариант E15

Выполнил:
студент
группы
РТ5-31Б
Молодцов
М.В.

Москва, 2024 г

Текст программы

```
class File:
    def __init__(self, file_id, name, size):
        self.file_id = file_id
        self.name = name
        self.size = size
        self.directories = []

    def add_directory(self, directory):
        self.directories.append(directory)

class Directory:
    def __init__(self, dir_id, name):
        self.dir_id = dir_id
        self.name = name
        self.files = []

    def add_file(self, file):
        self.files.append(file)
        file.add_directory(self)

    def average_file_size(self):
        if len(self.files) == 0:
            return 0
        total_size = sum(file.size for file in self.files)
        return round(total_size / len(self.files), 2)

def query_1(directories):
    return [
        (directory.name, [file.name for file in directory.files])
        for directory in directories if "Каталог файлов" in
directory.name
    ]

def query_2(directories):
    return sorted(
        [(directory.name, directory.average_file_size()) for directory
in directories],
        key=lambda x: x[1]
    )

def query_3(files):
    return [
        (file.name, [directory.name for directory in
file.directories])
        for file in files if file.name.lower().startswith('a')
    ]

def create_sample_data():
    file1 = File(1, "file1.txt", 10)
    file2 = File(2, "file2.txt", 20)
    file3 = File(3, "file3.txt", 30)
    file4 = File(4, "file4.txt", 25)
    file5 = File(5, "apple.txt", 15)
    dir1 = Directory(1, "Каталог файлов 1")
    dir2 = Directory(2, "Каталог файлов 2")
    dir3 = Directory(3, "Другой каталог")
    dir1.add_file(file1)
    dir1.add_file(file2)
```

```
dir2.add_file(file3)
dir2.add_file(file4)
dir3.add_file(file5)
return [dir1, dir2, dir3], [file1, file2, file3, file4, file5]
```

Модульное тестирование

```
import unittest
class TestFileManagement(unittest.TestCase):

    def setUp(self):
        """Создание тестовых данных для каждого теста."""
        self.directories, self.files = create_sample_data()

    def test_query_1(self):
        """Проверка запроса 1: Каталоги с 'Каталог файлов' в
названии."""
        result = query_1(self.directories)
        self.assertEqual(result, [
            ("Каталог файлов 1", ["file1.txt", "file2.txt"]),
            ("Каталог файлов 2", ["file3.txt", "file4.txt"])
        ])

    def test_query_2(self):
        """Проверка запроса 2: Каталоги с их средней
загруженностью."""
        result = query_2(self.directories)
        self.assertEqual(result, [
            ("Другой каталог", 15.0),
            ("Каталог файлов 1", 15.0),
            ("Каталог файлов 2", 25.0)
        ])

    def test_query_3(self):
        """Проверка запроса 3: Файлы, начинающиеся с 'A', и их
каталоги."""
        result = query_3(self.files)
        self.assertEqual(result, [
            ("apple.txt", ["Другой каталог"])
        ])

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