

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №9
дисциплины «Анализ данных»

Выполнил:
Степанов Леонид Викторович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение
средств вычислительной
техники и автоматизирование
систем», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Управление потоками в Python

Цель: приобретение навыков написания многопоточных приложений на языке программирования Python

Ход работы:

Индивидуальное задание (Вариант 1): С использованием многопоточности для заданного значения найти сумму ряда с точностью члена ряда по абсолютному значению и произвести сравнение полученной суммы с контрольным значением функции для двух бесконечных рядов.

Создал файл (ind.py) в котором было выполнено индивидуальное задание: создана функция многопоточного расчета элемента ряда (рис.1) с заданной точностью (рис.2), а также было произведено сравнение полученной суммы с контрольным значением (рис.3), результат выполнения для разных x представлен на рисунке 4.

```
def calculate_part(results, index, x, cur):  
    results[index] = 1  
  
    def my_log():  
        results[index] *= math.pow(math.log(3), cur)  
  
    def my_pow():  
        results[index] *= x**cur  
  
    def my_fact():  
        results[index] /= math.factorial(cur)  
  
    th1 = threading.Thread(target=my_log)  
    th2 = threading.Thread(target=my_pow)  
    th3 = threading.Thread(target=my_fact)  
  
    th1.start()  
    th2.start()  
    th3.start()
```

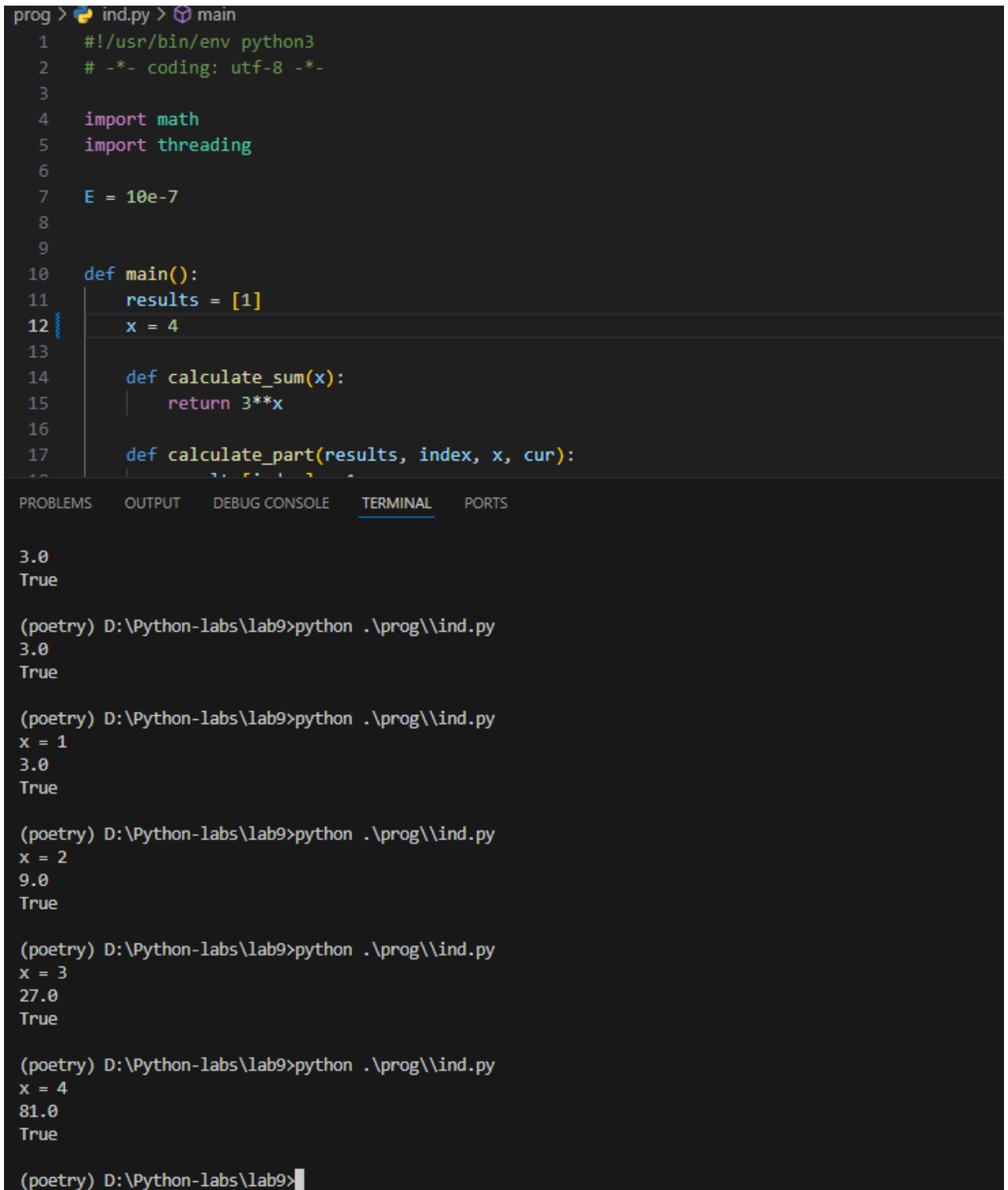
Рисунок 1 – Функция calculate_part

```
i = 0  
while results[i] > E:  
    results.append(0)  
    calculate_part(results, i + 1, x, i + 1)  
    i += 1
```

Рисунок 2 – Применение функции с точностью $10e-7$

```
print(round(sum(results), 5))
print(round(sum(results), 5) == calculate_sum(x))
```

Рисунок 3 – Сравнение с контрольным значением



```
prog > ind.py > main
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import math
5  import threading
6
7  E = 10e-7
8
9
10 def main():
11     results = [1]
12     x = 4
13
14     def calculate_sum(x):
15         return 3**x
16
17     def calculate_part(results, index, x, cur):
18         ...
19
20 if __name__ == '__main__':
21     main()
22
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
3.0
True

(poetry) D:\Python-labs\lab9>python .\prog\ind.py
3.0
True

(poetry) D:\Python-labs\lab9>python .\prog\ind.py
x = 1
3.0
True

(poetry) D:\Python-labs\lab9>python .\prog\ind.py
x = 2
9.0
True

(poetry) D:\Python-labs\lab9>python .\prog\ind.py
x = 3
27.0
True

(poetry) D:\Python-labs\lab9>python .\prog\ind.py
x = 4
81.0
True

(poetry) D:\Python-labs\lab9>
```

Рисунок 4 – Результат выполнения ind.py

Вывод: в ходе выполнения практической работы были приобретены навыки написания многопоточных приложений на языке программирования Python

