

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 ЛАБОРАТОРНОЙ РАБОТЕ №9
дисциплины «Анализ данных»
Вариант 29

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

(подпись)

Руководитель практики:
Воронкин Р.А., канд. технических
наук, 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 версии 3.x.

Порядок выполнения работы:

Вариант 29

Задание 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.

$$4. \quad S = \sum_{n=0}^{\infty} \left(\frac{1}{2^n} + \frac{1}{3^n} \right) x^{n-1} = \frac{5}{6} + \frac{13}{36}x + \frac{35}{216}x^2 + \dots; \quad x = -0,8; \quad y = \frac{5-2x}{6-5x+x^2}.$$

$$5. \quad S = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots; \quad x = 0,3; \quad y = \cos x.$$

Рисунок 1 – Условие задачи

Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

# С использованием многопоточности для
# заданного значения x найти сумму ряда S с
# точностью члена ряда по абсолютному
# значению  $\epsilon=10^{-7}$  и произвести сравнение
# полученной суммы
# с контрольным значением функции y
# для двух бесконечных рядов.
# Варианты 29 и 30 (4 и 5)

import math
from threading import Lock, Thread

lock = Lock()

# 4 (29) вариант
def sum_row_1(x, eps, s_dict):
    s = 0
    n = 1
    while True:
        a = 1/(2 ** n)
        b = 1/(3 ** n)
        c = math.pow(x, n-1)
        element = (a + b)*c

        if abs(element) < eps:
```

```

        break
    else:
        s += element
        n += 1
with lock:
    s_dict["row_1"] = s

# 5 (30) вариант
def sum_row_2(x, eps, sum_dict):
    sum = 0
    n = 0
    f = 1
    a = 2
    i = 1
    while True:
        z = a*i
        k = x**(2*n)
        f *= (z-1)*z
        element = ((-1)** n) * k / (f/2)
        i = n+1

        if abs(element) < eps:
            break
        else:
            sum += element
            n += 1
    with lock:
        sum_dict["row_2"] = sum

def main():
    sum_dict = { }

    eps = 10**-7
    # 4 Вариант
    x1 = -0.8
    y1 = (5-2*x1) / (6-5*x1+(x1**2))
    # 5 Вариант
    x2 = 0.3
    y2 = math.cos(x2)

    thread1 = Thread(target=sum_row_1, args=(x1, eps, sum_dict))
    thread2 = Thread(target=sum_row_2, args=(x2, eps, sum_dict))

    thread1.start()
    thread2.start()

    thread1.join()
    thread2.join()

    sum_1 = sum_dict["row_1"]
    sum_2 = sum_dict["row_2"]

```

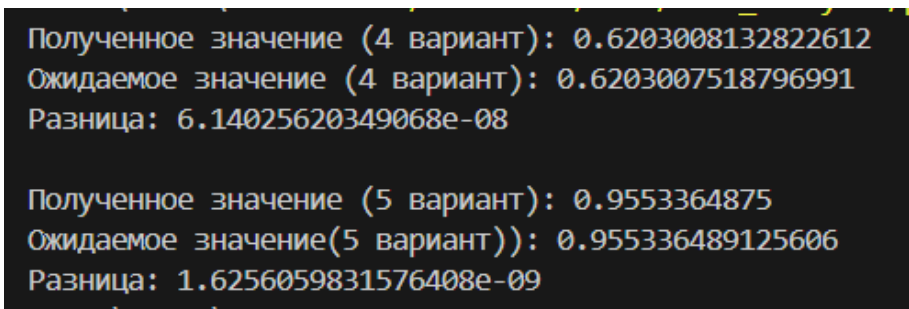
```

print(
    f"Полученное значение (4 вариант): {sum_1}"
    f"\nОжидаемое значение (4 вариант): {y1}"
    f"\nРазница: {abs(sum_1 - y1)}"
)
print(
    f"\nПолученное значение (5 вариант): {sum_2}"
    f"\nОжидаемое значение(5 вариант)): {y2}"
    f"\nРазница: {abs(sum_2 - y2)}"
)

if __name__ == "__main__":
    main()

```

Результат работы программы:



```

Полученное значение (4 вариант): 0.6203008132822612
Ожидаемое значение (4 вариант): 0.6203007518796991
Разница: 6.14025620349068e-08

Полученное значение (5 вариант): 0.9553364875
Ожидаемое значение(5 вариант)): 0.955336489125606
Разница: 1.6256059831576408e-09

```

Рисунок 2 – Результат работы программы

Выводы: в ходе выполнения заданий лабораторной работы получены навыки использования многопоточности в Python