

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 №11**  
**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 процессами в Python

Цель: приобретение навыков написания многозадачных приложений на языке программирования Python версии 3.x

Порядок выполнения работы:

Вариант 29

Задание 1. Для своего индивидуального задания лабораторной работы 2.23 необходимо реализовать вычисление значений в двух функций в отдельных процессах

Код программы:

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

# Для своего индивидуального задания
# лабораторной работы 2.23 необходимо реализовать
# вычисление значений в двух функций в отдельных
# процессах
# Варианты 29 и 30 (4 и 5)

import math
from multiprocessing import Barrier, Manager, Process

# 4 (29) вариант
def sum_row_1(x, eps, s_dict, br, lock):
    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
    br.wait()

# 5 (30) вариант
def sum_row_2(x, eps, sum_dict, br, lock):
    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
br.wait()

def conveyor(sum_dict, y1, y2, br):
    br.wait()
    sum_1 = sum_dict["row_1"]
    sum_2 = sum_dict["row_2"]
    print("Функция conveyor(): \n")
    print(
        f"Полученное значение (4 вариант): {sum_1}"
        f"\nОжидаемое значение (4 вариант): {y1}"
        f"\nРазница: {abs(sum_1 - y1)}"
    )
    print(
        f"Полученное значение (5 вариант): {sum_2}"
        f"\nОжидаемое значение(5 вариант): {y2}"
        f"\nРазница: {abs(sum_2 - y2)}"
    )

def main(manager):
    br = Barrier(3)
    lock = manager.Lock()
    sum_dict = manager.dict()

    eps = 10**-7

    x1 = -0.8
    y1 = (5-2*x1) / (6-5*x1+(x1**2))

    x2 = 0.3
    y2 = math.cos(x2)

    process1 = Process(target=sum_row_1, args=(x1, eps, sum_dict, br, lock))
    process2 = Process(target=sum_row_2, args=(x2, eps, sum_dict, br, lock))
    process3 = Process(target=conveyor, args=(sum_dict, y1, y2, br))

    process1.start()
    process2.start()

```

```
process3.start()

process1.join()
process2.join()
process3.join()
if __name__ == "__main__":
    with Manager() as manager:
        main(manager)
```

Результат работы программы:

Функция conveyor():

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  
PS C:\Users\HAIER>

Рисунок 1 – Результат работы программы

Выводы: в ходе выполнения заданий лабораторной работы получены навыки написания программ с использованием процессов.