

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 №11
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 процессами в Python

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

Ход работы:

Индивидуальное задание: для своего индивидуального задания лабораторной работы 2.23 необходимо реализовать вычисление значений в двух функций в отдельных процессах.

Создал файл (ind.py) в котором реализовал вычисление части ряда в разных процессах с использованием многопоточности, на рис. 1 изображена реализация появления процессов, на рис. 2 изображена функция calculate_part которая служит для передачи информации в основной интерпретатор и для многопоточного расчета.

```
def main():
    x = 3
    i = 0

    while not (round(sum(results), 5) == calculate_sum(x)):
        Process(target=calculate_part, args=(x, i + 1, local_results)).start()
        results.append(local_results.get())
        i += 1

    print(results)
    print(f"x = {x}")
    print(round(sum(results), 5))
    print(round(sum(results), 5) == calculate_sum(x))
```

Рисунок 1 – Функция main()

```
def calculate_part(x, cur, result_queue):
    local_result = [1]

    def my_log(local_result):
        local_result[0] *= math.pow(math.log(3), cur)
        br.wait()

    def my_pow(local_result):
        local_result[0] *= x**cur
        br.wait()

    def my_fact(local_result):
        local_result[0] /= math.factorial(cur)
        br.wait()

    Thread(target=my_log, args=(local_result,)).start()
    Thread(target=my_pow, args=(local_result,)).start()
    Thread(target=my_fact, args=(local_result,)).start()

    br.wait()
    result_queue.put(local_result[0])
```

Рисунок 2 – Функция calculate_part()

Вывод: в ходе выполнения практической работы были приобретены навыки написания многозадачных приложений на языке программирования Python версии 3.x.