

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»

Выполнил:

Пустяков Андрей Сергеевич

2 курс, группа ИВТ-б-о-22-1,

09.03.01 «Информатика и
вычислительная техника (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:

Воронкин Р. А. кандидат технических
наук, доцент, доцент кафедры
инфокоммуникаций

(подпись)

Ставрополь, 2024 г.

Тема: Управление процессами в Python.

Цель: приобрести навыки написания многозадачных приложений на языке программирования Python версии 3.x.

Ход работы:

Создание общедоступного репозитория на «GitHub», клонирование репозитория, редактирование файла «.gitignore», организация репозитория согласно модели ветвления «git-flow» (рис. 1).

```
C:\Users\Andrey\Desktop\Анализ_данных\11_лаба\Analysis_data_laboratory_work_11>git flow init
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]
How to name your supporting branch prefixes?
Feature branches? [feature/]
```

Рисунок 1 – Организация модели ветвления «git-flow»

Выполнение индивидуальных заданий:

Задание 1.

Необходимо с использованием многопоточности для заданного значения x найти сумму ряда S с точностью члена ряда по абсолютному значению и произвести сравнение полученной суммы с контрольным значением функции $y(x)$ для двух бесконечных рядов.

Необходимо доработать программу лабораторной работы 2.23, организовав вычисления значений двух функций в отдельных процессах.

Сумма ряда (Вариант 26 (1)):

1.
$$S = \sum_{n=0}^{\infty} \frac{x^n \ln^n 3}{n!} = 1 + \frac{x \ln 3}{1!} + \frac{x^2 \ln^2 3}{2!} + \dots; x = 1; y = 3^x.$$

Сумма ряда (Вариант 26 (2)):

2.
$$S = \sum_{n=0}^{\infty} x^n = 1 + x + x^2 + x^3 + \dots; x = 0,7; y = \frac{1}{1-x}.$$

Код программы данной задачи:

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

import math
from multiprocessing import Process, Manager
import sympy as sp

"""
Необходимо с использованием многопоточности для заданного значения x найти
сумму ряда S
с точностью члена ряда по абсолютному значению и произвести сравнение
полученной суммы с
контрольным значением функции y(x) для двух бесконечных рядов.
Необходимо доработать программу лабораторной работы 2.23, организовав
вычисления значений
двух функций в отдельных процессах.
Вариант 26, задачи 1 и 2.
"""

E = 1e-7 # Точность

def series_1(x, eps, results):
    """
    Функция вычисления суммы ряда задачи №1 (x = 1).
    """
    s = 0
    n = 0
    while True:
        term = x**n * sp.log(3)**n / math.factorial(n)
        if abs(term) < eps:
            break
        s += term
        n += 1
    results["series_1"] = s

def series_2(x, eps, results):
    """
    Функция вычисления суммы ряда задачи №2 (x = 0,7).
    """
    s = 0
    n = 0
    while True:
        term = x**n
        if abs(term) < eps:
            break
        s += term
        n += 1
    results["series_2"] = s

def main():
    """
    Главная функция программы.
    """
    with Manager() as manager:
        results = manager.dict({"series1": 0, "series2": 0})

        x1 = 1
        control_1 = 3**x1

        x2 = 0.7
        control_2 = 1/(1-x2)
```

```

# Создание процессов.
process_1 = Process(target=series_1, args=(x1, E, results))
process_2 = Process(target=series_2, args=(x2, E, results))

process_1.start()
process_2.start()

process_1.join()
process_2.join()

sum_1 = results["series_1"]
sum_2 = results["series_2"]

print(f"x1 = {x1}")
print(f"Сумма ряда 1: {sum_1:.7f}")
print(f"Контрольное значение 1: {control_1:.7f}")
print(f"Совпадение 1: {round(sum_1, 7) == round(control_1, 7)}")

print(f"x2 = {x2}")
print(f"Сумма ряда 2: {sum_2:.7f}")
print(f"Контрольное значение 2: {control_2:.7f}")
print(f"Совпадение 2: {round(sum_2, 7) == round(control_2, 7)}")

if __name__ == "__main__":
    main()

```

Результаты работы данной программы при заданных значениях x (рис. 2).

```

C:\Users\Andrey\anaconda3\envs\lab_9\python.exe C:\Users\Andrey\Desktop\Анализ_данных
x1 = 1
Сумма ряда 1: 2.9999999
Контрольное значение 1: 3.0000000
Совпадение 1: False
x2 = 0.7
Сумма ряда 2: 3.3333331
Контрольное значение 2: 3.3333333
Совпадение 2: False

```

Рисунок 2 – Результаты работы программы

Ответы на контрольные вопросы:

1. Как создаются и завершаются процессы в Python?

Для создания нового процесса в Python используется класс Process из модуля multiprocessing. Основные шаги: импортировать модуль multiprocessing, создать объект класса Process, передав целевую функцию и аргументы, запустить процесс с помощью метода start().

2. В чем особенность создания классов-наследников от Process?

Можно создать класс-наследник от Process для более сложных сценариев. Преимущество такого подхода в том, что вы можете переопределить метод `run()` для задания поведения процесса.

3. Как выполнить принудительное завершение процесса?

Для принудительного завершения процесса используется метод `terminate()`. Этот метод отправляет процессу сигнал завершения.

4. Что такое процессы-демоны? Как запустить процесс-демон?

Процесс-демон — это процесс, который работает в фоновом режиме и завершается, когда завершается основной процесс программы. Чтобы создать процесс-демон, нужно установить атрибут `daemon` объекта Process в `True` перед вызовом метода `start()`.

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