

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:
Говоров Егор Юрьевич
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 работы: приобретение навыков написания многозадачных приложений на языке программирования Python версии 3.x.

Ход работы

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.

2. Дополнил файл .gitignore необходимыми правилами.

3. Организовал созданный репозиторий в соответствии с необходимыми требованиями.

4. Добавил в файл README.md информацию о группе и ФИО студента, выполняющего лабораторную работу.

5. Выполнил индивидуальное задание. Привел в отчете скриншоты работы программы решения индивидуального задания.

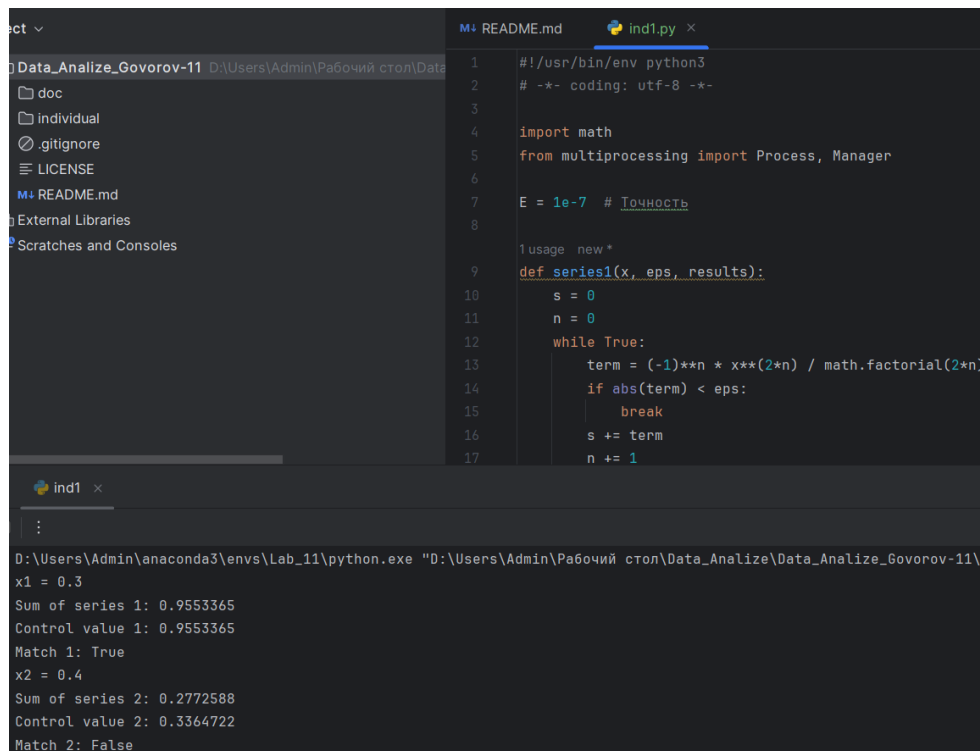
Для своего индивидуального задания лабораторной работы 2.23 необходимо реализовать вычисление значений в двух функций в отдельных процессах.

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

Рисунок 1. Функция варианта 5

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

Рисунок 2. Функция варианта 6



The screenshot shows a code editor with two tabs: 'README.md' and 'ind1.py'. The 'ind1.py' tab is active, displaying a Python script. The script imports 'math' and 'multiprocessing' (Process, Manager). It defines a constant 'E = 1e-7' and a function 'series1(x, eps, results)'. The function uses a 'while True' loop to calculate terms of a series until the absolute value of a term is less than 'eps'. The script also includes a 'usage' section and a 'def' statement for 'series1'. Below the code editor, a terminal window shows the execution of the script. The terminal output displays the sum of series 1, control value 1, match 1, x2, sum of series 2, control value 2, and match 2.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import math
5 from multiprocessing import Process, Manager
6
7 E = 1e-7 # Точность
8
9 usage new *
10 def series1(x, eps, results):
11     s = 0
12     n = 0
13     while True:
14         term = (-1)**n * x**(2*n) / math.factorial(2*n)
15         if abs(term) < eps:
16             break
17         s += term
18         n += 1
```

```
D:\Users\Admin\anaconda3\envs\Lab_11\python.exe "D:\Users\Admin\Рабочий стол\Data_Analyze\Data_Analyze_Govorov-11\
x1 = 0.3
Sum of series 1: 0.9553365
Control value 1: 0.9553365
Match 1: True
x2 = 0.4
Sum of series 2: 0.2772588
Control value 2: 0.3364722
Match 2: False
```

Рисунок 3. Результат работы программы индивидуального задания

Контрольные вопросы

1. Как создаются и завершаются процессы в Python?

Классом, который отвечает за создание и управление процессами является `Process` из пакета `multiprocessing`. Он совместим по сигнатурам методов и конструктора с `threading.Thread`, это сделано для более простого перехода от многопоточкового приложения к многопроцессному. Помимо одноименных с `Thread` методов, класс `Process` дополнительно предоставляет ряд своих.

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

В классе наследнике от `Process` необходимо переопределить метод `run()` для того, чтобы он (класс) соответствовал протоколу работы с процессами.

3. Как выполнить принудительное завершение процесса?

В отличие от потоков, работу процессов можно принудительно завершить, для этого класс `Process` предоставляет набор методов: `terminate()` - принудительно завершает работу процесса. В Unix отправляется команда `SIGTERM`, в Windows используется функция

`TerminateProcess()`.

`kill()` - метод аналогичный `terminate()` по функционалу, только вместо `SIGTERM` в Unix будет отправлена команда `SIGKILL`.

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

Процессы демоны по своим свойствам похожи на потоки-демоны, их суть заключается в том, что они завершают свою работу, если завершился родительский процесс. Указание на то, что процесс является демоном должно быть сделано до его запуска (до вызова метода `start()`). Для демонического процесса запрещено самостоятельно создавать дочерние процессы. Эти процессы не являются демонами (сервисами) в понимании Unix, единственное их свойство – это завершение работы вместе с родительским процессом.

Указать на то, что процесс является демоном можно при создании экземпляра класса через аргумент `daemon`, либо после создания через свойство `daemon`.

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