

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Лабораторная работа 2.25 Управление процессами в Python»

ОТЧЕТ
по лабораторной работе №28
дисциплины
«Основы программной инженерии»

Выполнил:

Луценко Дмитрий Андреевич
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Лабораторная работа 2.25 Управление процессами в Python

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

Ход работы:

. Разработать приложение, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) с помощью паттерна “Производитель-Потребитель”, условие которой предварительно необходимо согласовать с преподавателем.

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

from math import log, sqrt
from multiprocessing import Process

EPS = .0000001

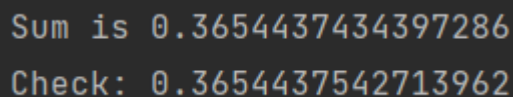
def sum_func(x):
    summ = x
    prev = 0
    i = 1
    while abs(summ - prev) > EPS:
        prev = summ
        summ += x ** (i * 2 + 1) / (i * 2 + 1)
        i += 1

    print(f"Sum is {summ}")

def check_func(x):
    res = log(sqrt((1 + x) / (1 - x)))
    print(f"Check: {res}")

if __name__ == '__main__':
    x = 0.35
    proc1 = Process(target=sum_func, args=(x,))
    proc2 = Process(target=check_func, args=(x,))
    proc1.start()
    proc2.start()
```

Листинг 1 – Код программы индивидуального задания



```
Sum is 0.3654437434397286
Check: 0.3654437542713962
```

Рисунок 1 – Результат выполнения программы

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

Ответы на контрольные вопросы:

1. Как создаются и завершаются процессы в Python? Процессы в Python создаются с помощью модуля multiprocessing. Для создания процесса необходимо создать объект класса Process, передав в конструктор функцию или метод класса, который будет выполняться в процессе. Для запуска процесса используется метод start(). Завершение процесса происходит автоматически при завершении выполнения функции или метода, переданного в конструктор Process.

2. В чем особенность создания классов-наследников от Process? Особенность создания классов-наследников от Process заключается в том, что в методе run() необходимо указать код, который будет выполняться в процессе. Этот метод вызывается при запуске процесса. Также класс наследник должен иметь свой конструктор, в котором нужно вызвать конструктор родительского класса и передать в него аргументы.

3. Как выполнить принудительное завершение процесса? Для принудительного завершения процесса в Python можно использовать метод terminate(), который отправляет процессу сигнал SIGTERM. Обработка этого сигнала внутри процесса приводит к его завершению. Однако, не рекомендуется использовать terminate() без необходимости, так как это может привести к утечкам ресурсов.

4. Что такое процессы-демоны? Как запустить процесс-демон? Процессы-демоны – это процессы, которые работают в фоновом режиме, не взаимодействуя с пользователем. Они не имеют своего терминала и не могут взаимодействовать с консолью. Запустить процесс-демон можно, используя методы из модуля daemonize. Кроме того, при создании процесса можно указать параметр daemon=True, что превращает процесс в демон.