

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.23

Работа с переменными окружения в Python3

по дисциплине «Управление потоками в Python»

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2022

Цель работы: приобретение навыков с управлением потоками с помощью языка программирования Python версии 3.x.

Ход работы: Примеры

<https://github.com/Mirror-Shard/L2.23>

1. Изучил теоретический материал и приступил к выполнению примеров:

```
1  from threading import Thread
2  from time import sleep
3
4
5  def func():
6      for i in range(5):
7          print(f"from child thread: {i}")
8          sleep(0.5)
9
10
11  th = Thread(target=func)
12  th.start()
13
14  for i in range(5):
15      print(f"from main thread: {i}")
16      sleep(1)
17
```

Рисунок 1 – Код примера

```
from child thread: 0from main thread: 0

from child thread: 1
from main thread: 1
from child thread: 2
from child thread: 3
from main thread: 2
from child thread: 4
from main thread: 3
from main thread: 4
```

Рисунок 2 – Результат работы примера

Задание 1

С использованием многопоточности для заданного значения найти сумму ряда с точностью члена ряда по абсолютному значению и произвести сравнение полученной суммы с контрольным значением функции для двух бесконечных рядов.

1. Написал программу для вычисления суммы ряда указанной точности с использованием многопоточности:

```
"C:/Users/1/Desktop/Алгоритмизация/Лабораторная 2.23/ind_1.py"
Результат сравнения: -1.0026259652063807
Результат сравнения: -1.3646626709935

Process finished with exit code 0
```

Рисунок 3 – Результат выполнения тестов

Контрольные вопросы:

1. Что такое синхронность и асинхронность?

Синхронное выполнение программы подразумевает последовательное выполнение операций. Асинхронное – предполагает возможность независимого выполнения задач.

2. Что такое параллелизм и конкурентность?

Конкурентность предполагает выполнение нескольких задач одним исполнителем. Из примера с готовкой: один человек варит картошку и

прибирается, при этом, в процессе, он может переключаться: немного прибрался, пошел помешал-посмотрел на картошку, и делает он это до тех пор, пока все не будет готово.

Параллельность предполагает параллельное выполнение задач разными исполнителями: один человек занимается готовкой, другой приборкой.

3. Что такое GIL? Какое ограничение накладывает GIL?

GIL — это аббревиатура от Global Interpreter Lock – глобальная блокировка интерпретатора. Он является элементом эталонной реализации языка Python, которая носит название CPython. Суть GIL заключается в том, что выполнять байт код может только один поток. Это нужно для того, чтобы упростить работу с памятью (на уровне интерпретатора) и сделать комфортной разработку модулей на языке C.

4. Каково назначение класса Thread?

За создание, управление и мониторинг потоков отвечает класс Thread из модуля threading. Поток можно создать на базе функции, либо реализовать свой класс – наследник Thread и переопределить в нем метод run().

5. Как реализовать в одном потоке ожидание завершения другого потока?

Если необходимо дождаться завершения работы потока перед тем как начать выполнять какую-то другую работу, то воспользуйтесь методом join().

6. Как проверить факт выполнения потоком некоторой работы?

Для того, чтобы определить выполняет ли поток какую-то работу или завершился используется метод is_alive().

7. Как реализовать приостановку выполнения потока на некоторый промежуток времени? У метода join() есть параметр timeout, через который задается время ожидания завершения работы потоков.

8. Как реализовать принудительное завершение потока?

В Python у объектов класса Thread нет методов для принудительного завершения работы потока. Один из вариантов решения этой задачи – это создать специальный флаг, через который потоку будет передаваться сигнал

остановки. Доступ к такому флагу должен управляться объектом синхронизации.

9. Что такое потоки-демоны? Как создать поток-демон? Поток демона – это тип потока, который может работать независимо в фоновом режиме. Эти типы потоков выполняются независимо от основного потока. Поэтому они называются неблокирующими потоками. Чтобы создать такой поток необходимо при создании объекта Thread аргументу `daemon` присвоить значение `True`, либо после создания потока, перед его запуском присвоить свойству `daemon` значение `True`.

Вывод: в ходе работы приобрёл навыки по работе с управлением потоками при написании программ с помощью языка программирования Python версии 3.