

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

Управление потоками в Python

По дисциплине «Теории программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Плотников Д. В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р. А. _____

(подпись)

Ставрополь 2022

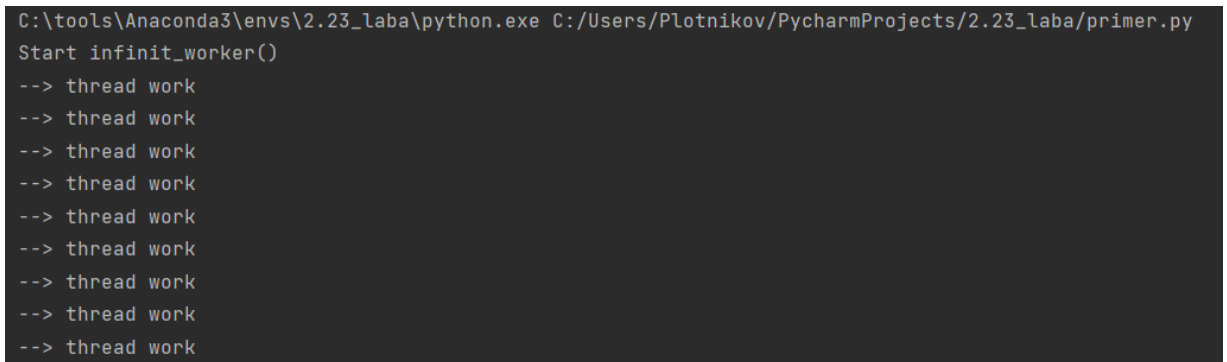
Цель работы: приобретение навыков написания многопоточных приложений на языке программирования Python версии 3.x.

Ход работы

1. Создал новый собственный репозиторий. Ссылка на репозиторий: https://github.com/Dmitry-15/2.23_laba.
2. С помощью команды `git clone` клонировал удаленный репозиторий на свой ПК. Дополнил файл `.gitignore` необходимыми правилами для работы с IDE PyCharm.
3. После ознакомления с теоретическим материалом приступил к выполнению заданий.

Пример 1

1. Для начала выполнил пример.



```
C:\tools\Anaconda3\envs\2.23_laba\python.exe C:/Users/Plotnikov/PycharmProjects/2.23_laba/primer.py
Start infinit_worker()
--> thread work
--> thread work
--> thread work
--> thread work
--> thread work
--> thread work
--> thread work
--> thread work
--> thread work
--> thread work
```

Рисунок 1 – Результат выполнения примера

2. Приступил к выполнению индивидуального задания.

Индивидуальное задание

Вариант 18

1. Условие задания: 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.

18.

$$S = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1} = x + \frac{x^3}{3} + \frac{x^5}{5} + \dots; \quad x = 0,35; \quad y = \ln \sqrt{\frac{1+x}{1-x}}.$$

```
C:\tools\Anaconda3\python.exe C:/Users/Plotnikov/2.23_laba/Tasks/individ1.py
Результат сравнения -2.8571428583155862
Результат сравнения -5.915392620181592e-10

Process finished with exit code 0
```

Рисунок 2 – Результат выполнения индивидуального задания

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 выполнения лабораторной работы успешно приобретены навыки написания многопоточных приложений на языке программирования Python версии 3.x.