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 РАБОТЕ №2.23
дисциплины «Программирование на языке Python»

Выполнил:
Кондратенко Даниил Витальевич
2 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные технологии и системы связи, очная форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р.А., канд. тех. наук, доцент,
доцент кафедры инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Тема: Управление потоками в Python.

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

Порядок выполнения работы:

Задание 1.

Изучил теоретический материал работы, создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами.

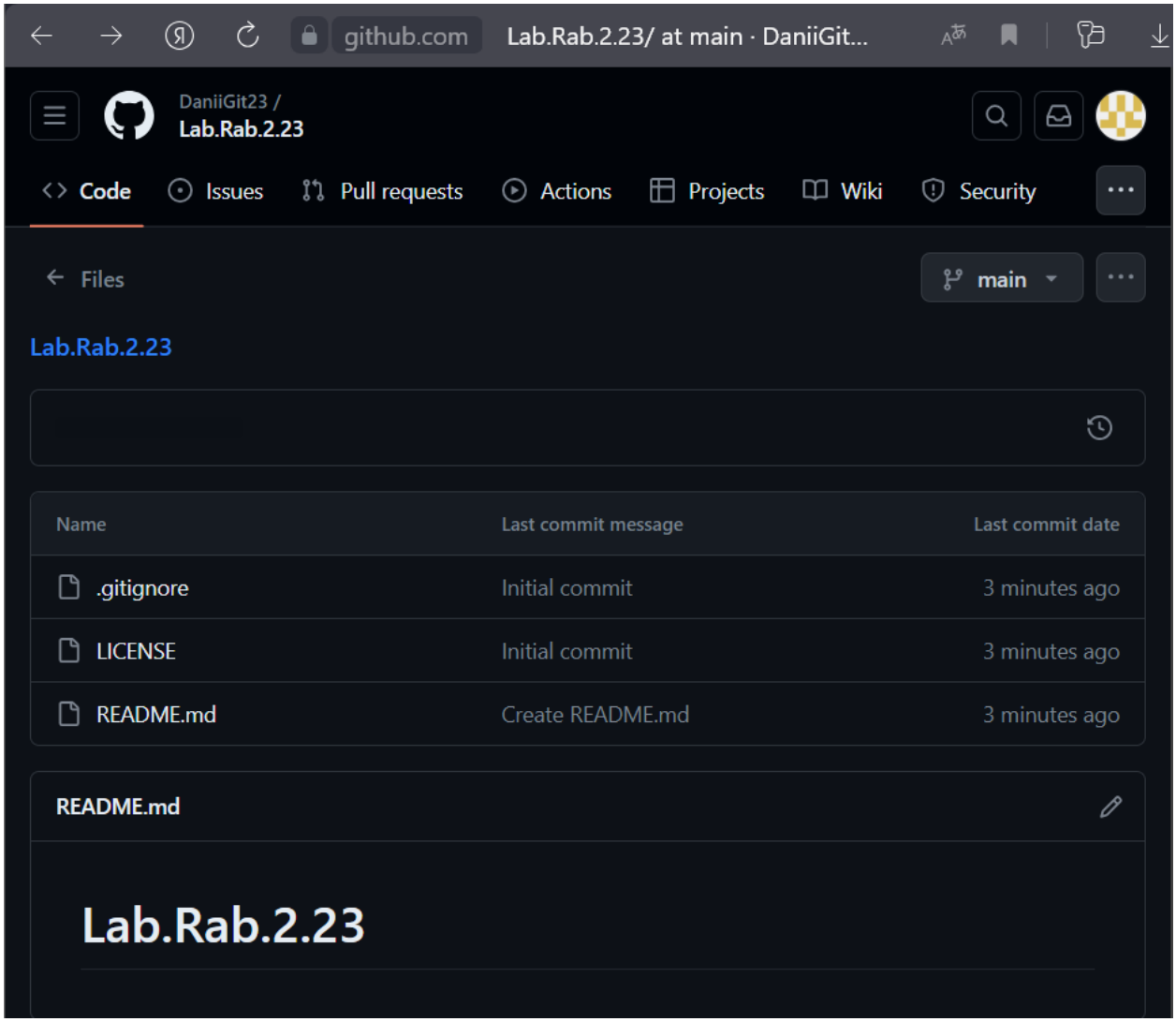

Рисунок 1. Новый репозиторий

Задание 2.

Проклонировал свой репозиторий на свой компьютер.

Организовал свой репозиторий в соответствие с моделью ветвления git-flow, появилась новая ветка develop.

```
C:\Users\HUAWEI>git clone https://github.com/DaniiGit23/Lab.Rab.2.23.git
Cloning into 'Lab.Rab.2.23'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.

C:\Users\HUAWEI>cd C:\Users\HUAWEI\Lab.Rab.2.23

C:\Users\HUAWEI\Lab.Rab.2.23>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/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/HUAWEI/Lab.Rab.2.23/.git/hooks]

C:\Users\HUAWEI\Lab.Rab.2.23>
```

Рисунок 2. Клонирование и модель ветвления git-flow

Реализовывал примеры и индивидуальные задания на основе ветки develop, без создания дополнительной ветки feature/(название ветки) по указанию преподавателя.

Задание 3.

Создал виртуальное окружение (ВО) Miniconda и активировал его, также установил необходимые пакеты isort, black, flake8.

```
(base) PS C:\Users\HUAWEI> cd C:\Users\HUAWEI\Lab.Rab.2.23
(base) PS C:\Users\HUAWEI\Lab.Rab.2.23> conda create -n 2.23 python=3.11
Retrieving notices: ...working... done
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Рисунок 3. Создание ВО

```
(base) PS C:\Users\HUAWEI\Lab.Rab.2.23> conda activate 2.23
(2.23) PS C:\Users\HUAWEI\Lab.Rab.2.23> conda install -c conda-forge black
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 23.5.2
  latest version: 23.10.0

Please update conda by running
```

Рисунок 4. Установка пакета black

```
(2.23) PS C:\Users\HUAWEI\Lab.Rab.2.23> conda install -c conda-forge flake8
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 23.5.2
  latest version: 23.10.0

Please update conda by running

    $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

     conda install conda=23.10.0
```

Рисунок 5. Установка пакета flake8

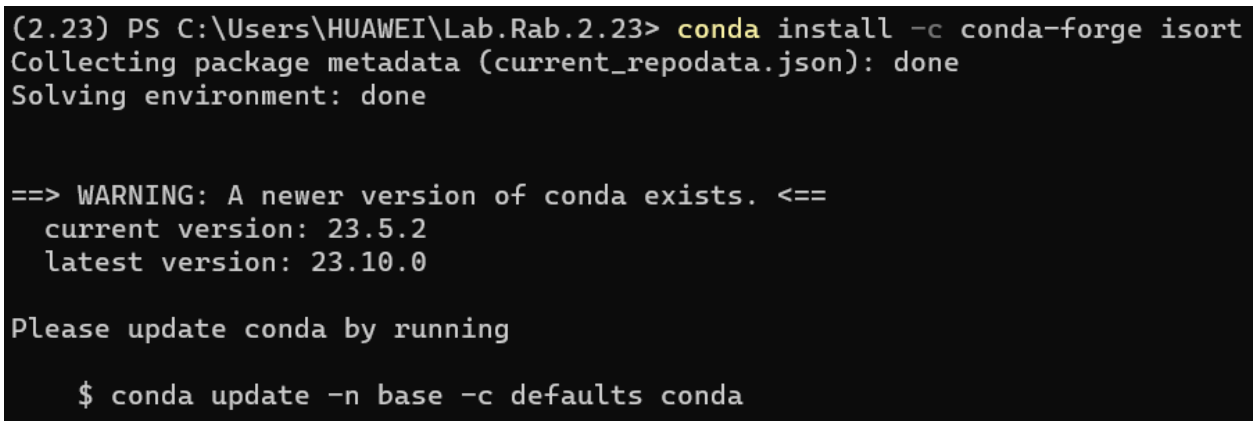

Рисунок 6. Установка пакета isort

Пакет isort (isrot) является инструментом для автоматической сортировки импортов в Python-кодах. Он используется для удобства чтения и поддержания порядка в коде.

Пакет black представляет инструмент автоматического форматирования кода для языка Python. Он помогает обеспечить единообразие стиля кодирования в проекте и улучшает читаемость кода.

Пакет flake8 отвечает за статический анализ и проверку Python-кода. Он проводит проверку на соответствие стилю кодирования PEP 8, а также наличие потенциальных ошибок и проблемных паттернов в коде.

Задание 3.

Выполнение индивидуального задания.

Условие задания: с использованием многопоточности для заданного значения **x** найти сумму ряда **S** с точностью члена ряда по абсолютному значению e = 10^-7 и произвести сравнение полученной суммы с контрольным значением функции для двух бесконечных рядов.

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

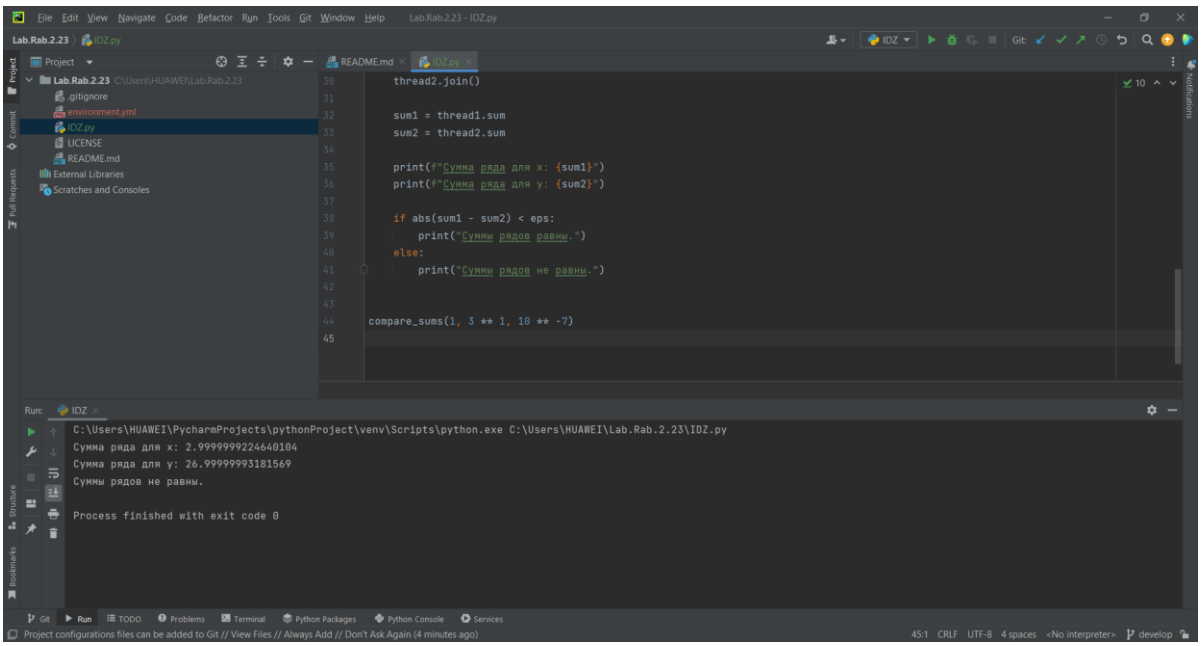

Рисунок 7. Результат индивидуального задания

Код начинается с импорта необходимых модулей - `math` для математических операций и `threading` для работы с многопоточностью.

Следующим определяется класс `SumThread`, который наследуется от `threading.Thread`. Этот класс переопределяет метод `__init__`, чтобы принимать два аргумента: `x` и `eps`. В методе `__init__` также инициализируется переменная `self.sum`, которая будет хранить вычисленную сумму.

Метод `run` вычисляет члены ряда и добавляет их к `self.sum`, пока абсолютное значение члена ряда больше `eps`.

Следующим определяется функция `compare_sums`, которая создает два экземпляра класса `SumThread`, каждый из которых вычисляет сумму ряда для

своего аргумента. Затем она запускает оба потока с помощью метода `start`, а затем ожидает их завершения с помощью метода `join`.

После завершения работы потоков, функция `compare_sums` извлекает вычисленные суммы из потоков и выводит их на экран. Затем она сравнивает эти суммы и выводит сообщение о том, равны ли они.

Наконец, вызывается функция `compare_sums` с аргументами `1`, `3**1` (то же самое, что и `3`) и `10**-7`.

Ссылка на репозиторий: https://github.com/DaniiGit23/Lab.Rab.2.23

Ответы на контрольные вопросы:

1. Синхронность и асинхронность: Синхронность - это последовательное выполнение операций. Операции выполняются последовательно, одна за другой. Асинхронность - это параллельное выполнение операций. Операции выполняются одновременно, не дожидаясь завершения предыдущих операций.

2. Параллелизм и конкурентность: Параллелизм - это выполнение нескольких задач одновременно. Конкурентность - это выполнение нескольких задач, где результат зависит от порядка их выполнения.

3. GIL (Global Interpreter Lock): GIL - это механизм синхронизации, который позволяет выполнять только один поток Python в один момент времени, чтобы предотвратить "гонки" между потоками.

4. Класс Thread: Класс Thread используется для создания нового потока.

5. Ожидание завершения другого потока: Для ожидания завершения другого потока можно использовать метод join().

6. Проверка факта выполнения потоком некоторой работы: Для проверки факта выполнения потоком некоторой работы можно использовать метод is_alive().

7. Приостановка выполнения потока на некоторый промежуток времени: Для приостановки выполнения потока на некоторый промежуток времени можно использовать метод sleep().

8. Принудительное завершение потока: Для принудительного завершения потока можно использовать метод stop(). Однако, этот метод устарел и не рекомендуется к использованию.

9. Потоки-демоны: Потоки-демоны - это потоки, которые запускаются в фоновом режиме и не препятствуют завершению работы программы. Для создания потока-демона можно использовать метод setDaemon(True).

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