

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 №2.24
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: синхронизация потоков в языке программирования Python.

Цель работы: приобретение навыков использования примитивов синхронизации в 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.

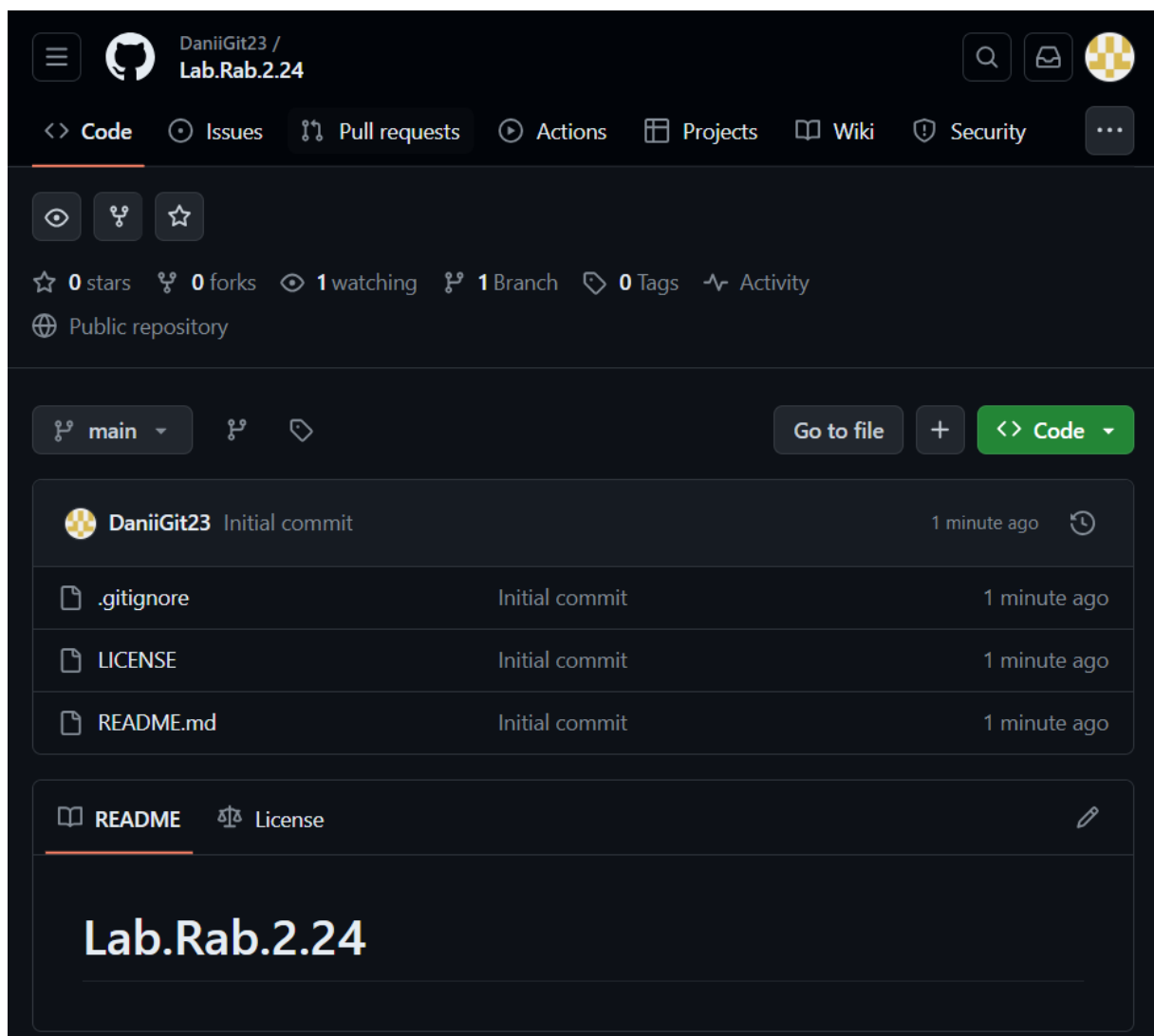


Рисунок 1. Новый репозиторий

Задание 2.

Проклонировал свой репозиторий на свой компьютер.

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

```
C:\Users\HUAWEI>git clone https://github.com/DaniiGit23/Lab.Rab.2.24.git
Cloning into 'Lab.Rab.2.24'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\HUAWEI>cd C:\Users\HUAWEI\Lab.Rab.2.24

C:\Users\HUAWEI\Lab.Rab.2.24>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.2.24/.git/hooks]

C:\Users\HUAWEI\Lab.Rab.2.24>
```

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.2.24  
(base) PS C:\Users\HUAWEI\Lab.Rab.2.24> conda create -n 2.24 python=3.11  
Retrieving notices: ...working... done  
Collecting package metadata (current_repodata.json): done  
Solving environment: done
```

Рисунок 3. Создание ВО

```
(2.24) PS C:\Users\HUAWEI\Lab.Rab.2.24> 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: 23.11.0
```

Рисунок 4. Установка пакета black

```
(2.24) PS C:\Users\HUAWEI\Lab.Rab.2.24> 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: 23.11.0
```

Рисунок 5. Установка пакета flake8

```
(2.24) PS C:\Users\HUAWEI\Lab.Rab.2.24> conda install -c conda-forge isort  
Collecting package metadata (current_repodata.json): done  
Solving environment: done  
  
==> WARNING: A newer version of conda exists. <==  
current version: 23.5.2  
latest version: 23.11.0
```

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 1.

Выполнение примера №1.

Пример работы с условной переменной.

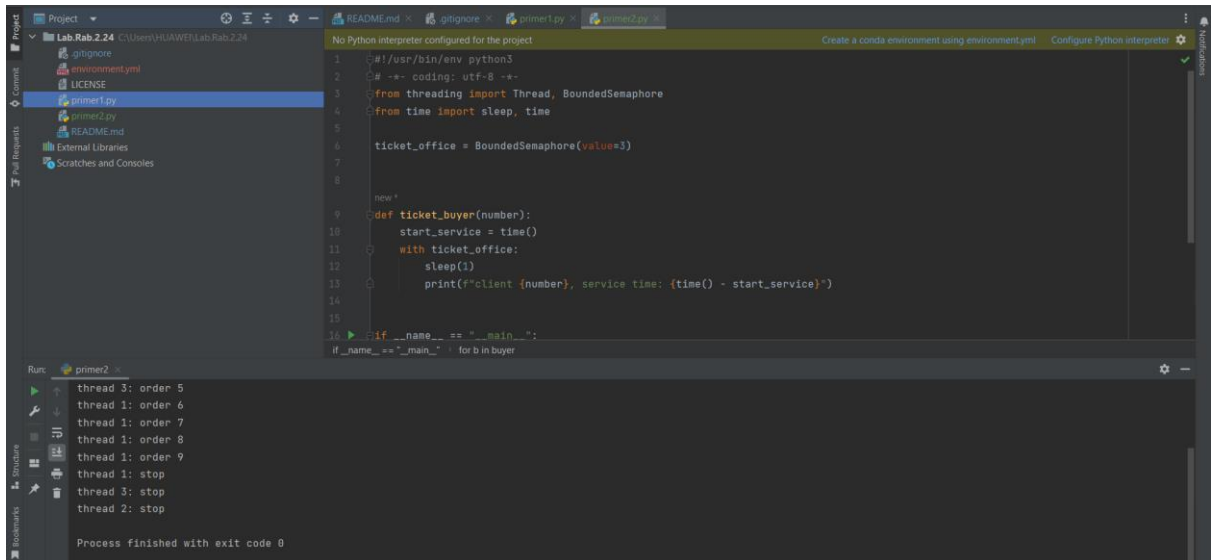


Рисунок 7. Работа с условной переменной

Выполнение примера №2.

В качестве примера приведем программу, моделирующую продажу билетов: обслуживание одного клиента занимает одну секунду, касс всего три, клиентов пять.

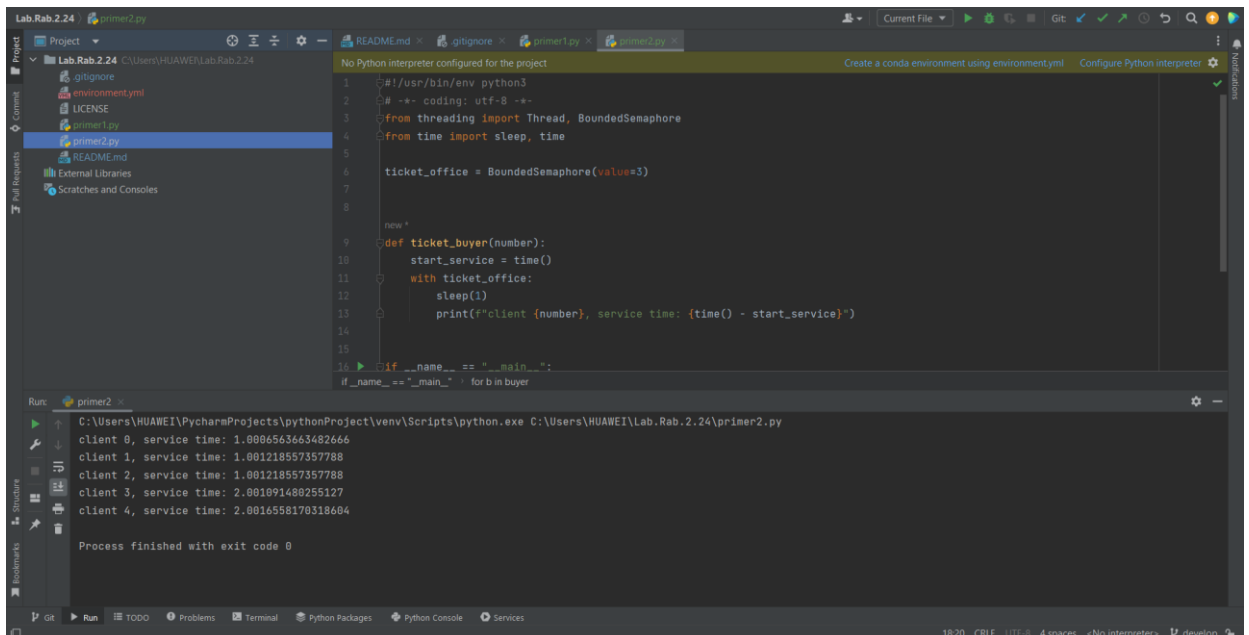


Рисунок 8. Программа, моделирующая продажу билетов

Выполнение примера №3.

Пример работы с Event-объектом:

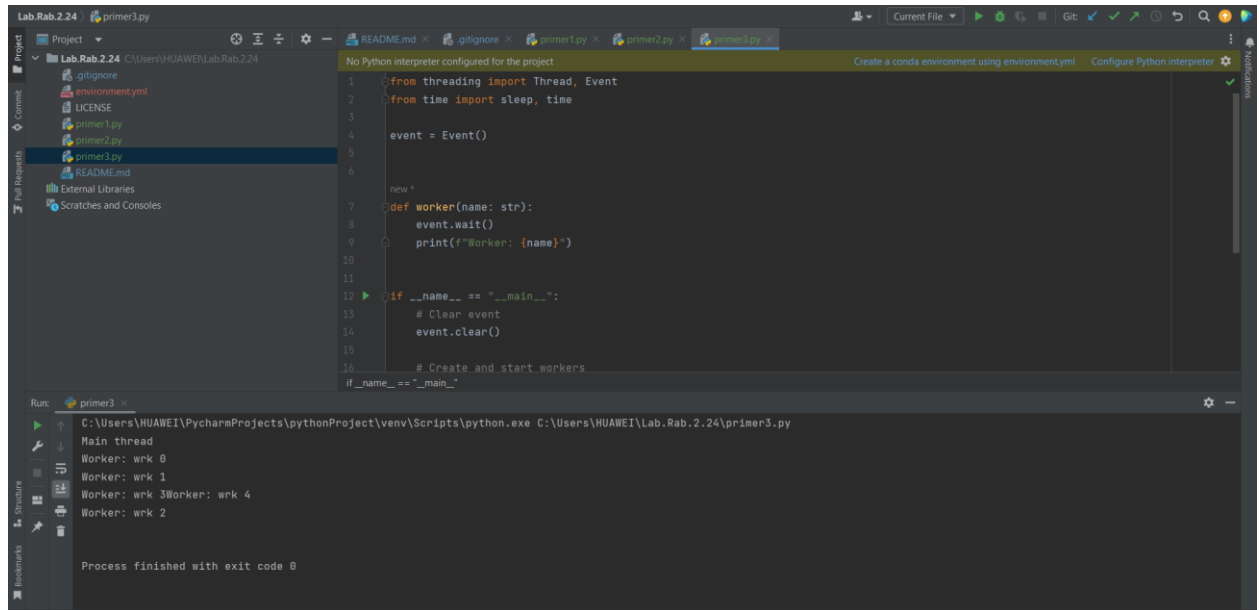


Рисунок 9. Работа с Event-объектом

Выполнение пример №4.

Пример работы с классом Barrier:

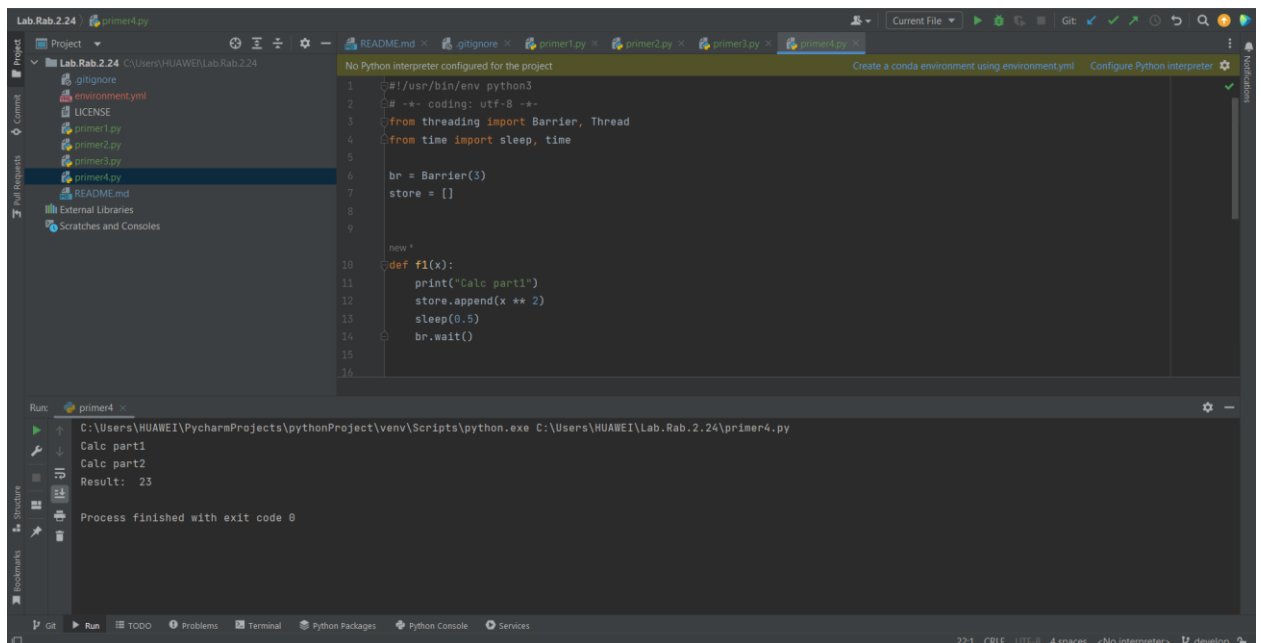


Рисунок 10. Работа с классом Barrier

Задание 2.

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 задания: в программе будет два потока - производитель и потребитель. Производитель будет генерировать числа от 1 до N, а потребитель будет суммировать квадраты этих чисел.

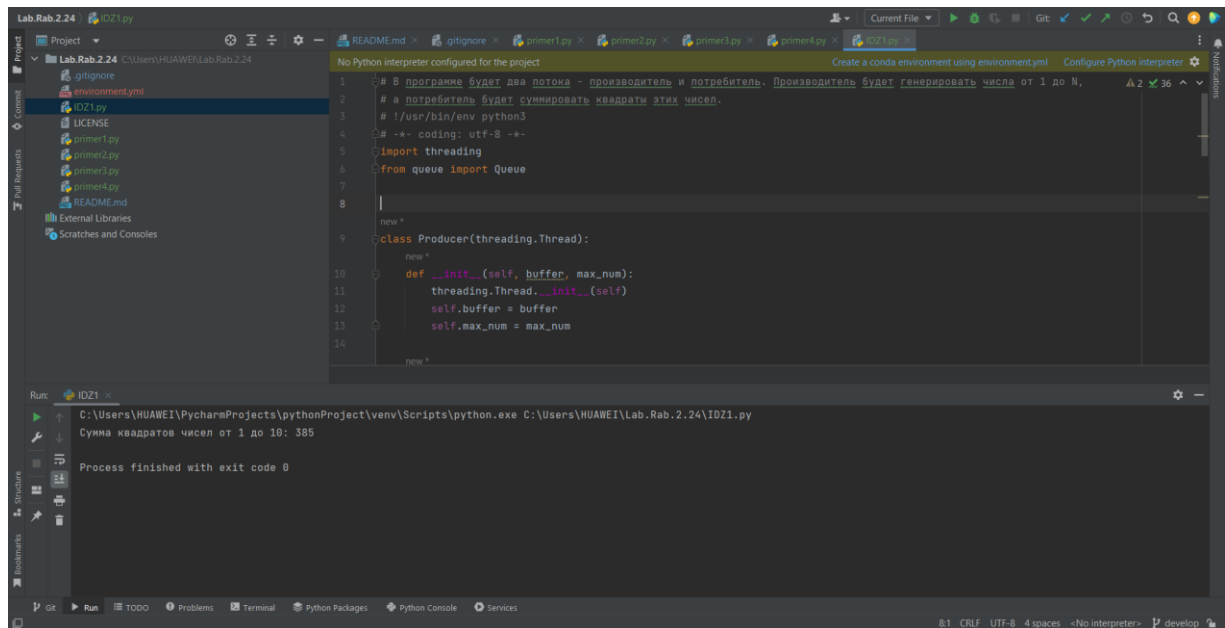


Рисунок 11. Выполнение индивидуального задания

Задание 3.

Для своего индивидуального задания лабораторной работы 2.23 необходимо организовать конвейер, в котором сначала в отдельном потоке вычисляется значение первой функции, после чего результаты вычисления должны передаваться второй функции, вычисляемой в отдельном потоке. Потоки для вычисления значений двух функций должны запускаться одновременно.

Условие задания: с использованием многопоточности для заданного значения x найти сумму ряда S с точностью члена ряда по абсолютному

значению $\epsilon = 10^{-7}$ и произвести сравнение полученной суммы с контрольным значением функции для двух бесконечных рядов.

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

Для организации конвейера, будем использовать класс Queue из модуля queue. В нем будем передавать результаты вычисления первой функции и получать их во второй функции.

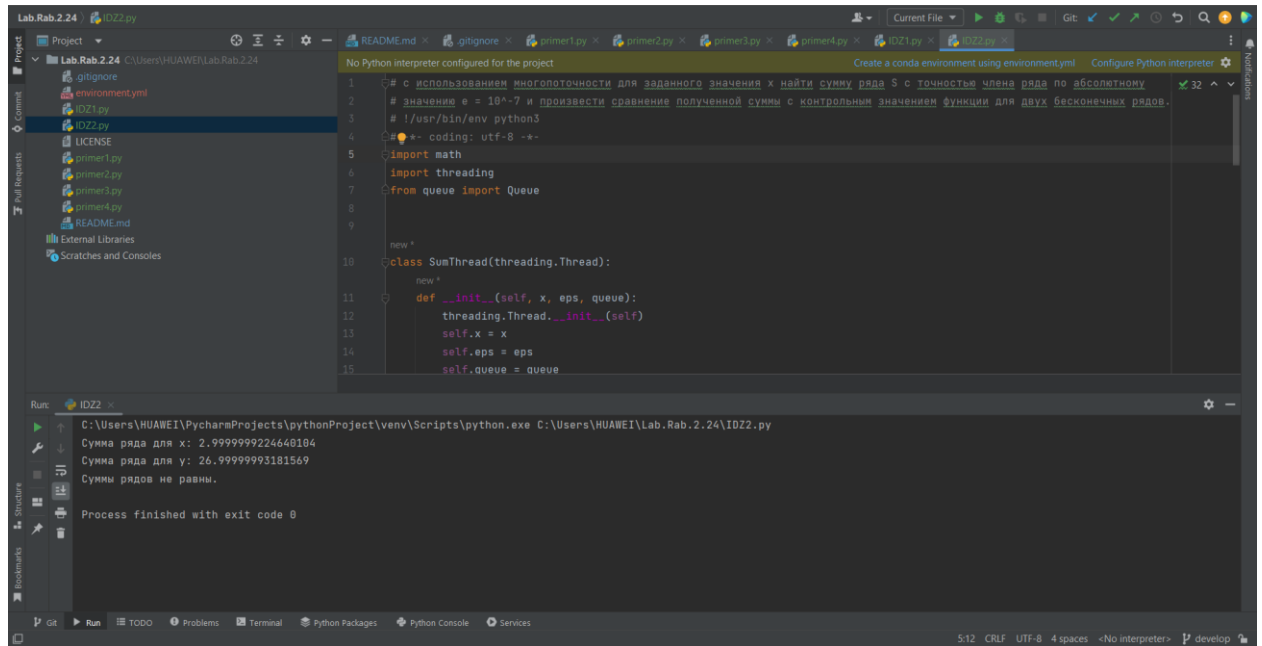


Рисунок 12. Выполнение индивидуального задания

Ответы на контрольные вопросы:

1. Lock-объекты в Python используются для синхронизации доступа к общим ресурсам в многопоточных программах. Они позволяют только одному потоку выполнять определенный блок кода в любой момент времени.

Приемы работы с Lock-объектом:

- `acquire()` - захватывает блокировку. Если блокировка занята, поток будет ждать, пока она освободится.
- `release()` - освобождает блокировку.

2. RLock-объекты (reentrant lock) - это вид Lock-объектов, которые могут быть захвачены несколько раз, считая поток владельцем блокировки.

Это позволяет потоку повторно захватывать блокировку, если он уже ее захватил.

3. Условные переменные используются для синхронизации потоков. Они позволяют потокам ждать, пока не будет выполнено определенное условие.

Методы условных переменных:

- `wait()` - поток ждет, пока условие не станет истинным.
- `notify()` - уведомляет одного из потоков, которые ждут это условие.
- `notify_all()` - уведомляет все потоки, которые ждут это условие.

4. Семафор - это объект синхронизации, который позволяет ограничить доступ к ресурсу определенным количеством потоков.

5. Событие - это объект синхронизации, который позволяет потокам ждать, пока не произойдет определенное событие.

6. Таймер - это объект синхронизации, который позволяет запланировать выполнение определенного кода в будущем.

7. Барьер - это объект синхронизации, который позволяет нескольким потокам синхронизироваться между собой.

8. Вывод: выбор примитива синхронизации зависит от конкретной задачи. Некоторые примитивы более эффективны, чем другие, и выбор зависит от конкретных требований к параллелизму и синхронизации.

Вывод: в ходе выполнения лабораторной работы приобрели навыки использования примитивов синхронизации в языке программирования Python версии 3.x.