

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 №2.10
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Кондратенко Даниил Витальевич
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», 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: функции с переменным числом параметров в Python.

Цель работы: приобретение навыков по работе с функциями с переменным числом параметров при написании программ с помощью языка 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.

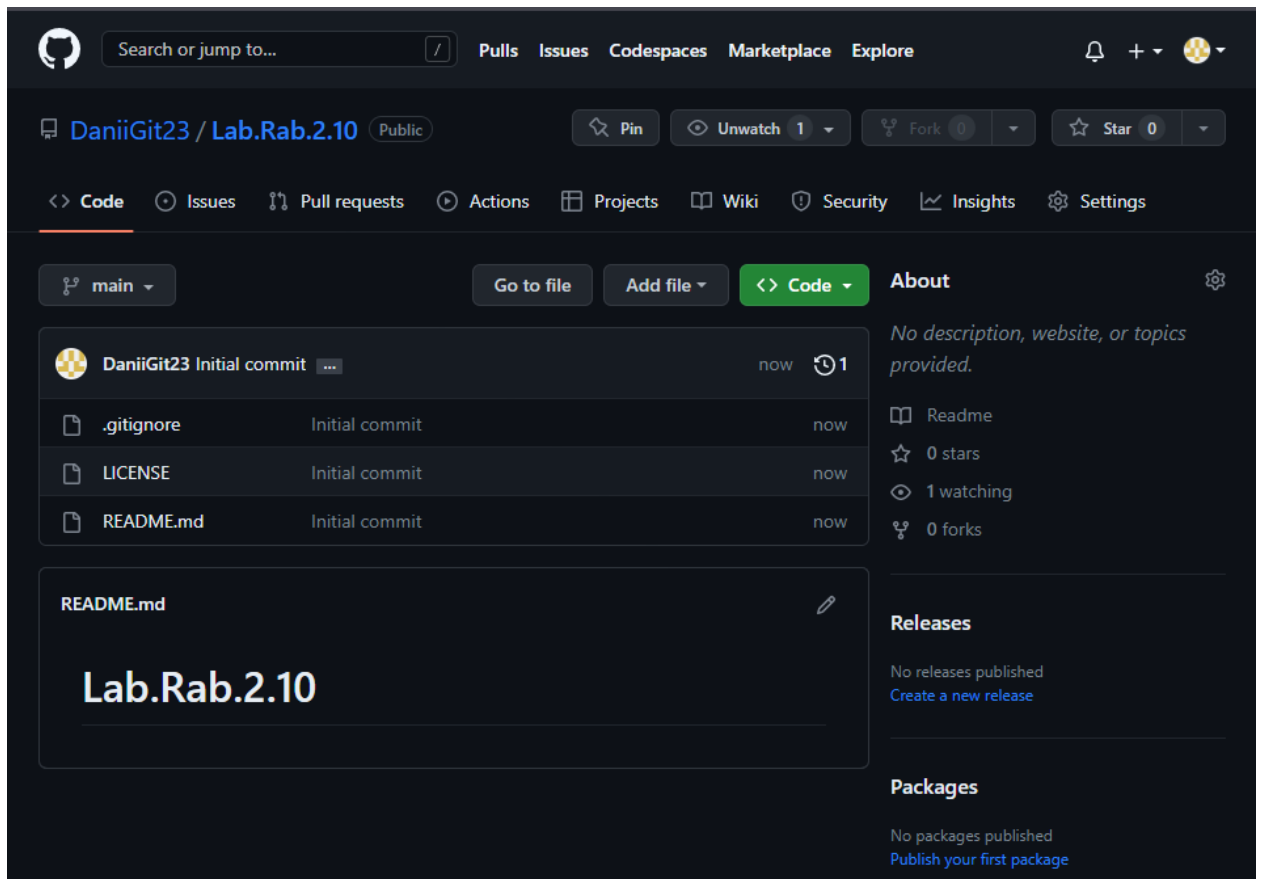


Рисунок 1 – Новый репозиторий

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.2.10.git
Cloning into 'Lab.Rab.2.10'...
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.2.10

C:\Users\HUAWEI\Lab.Rab.2.10>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

C:\Users\HUAWEI\Lab.Rab.2.10>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/]
```

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.

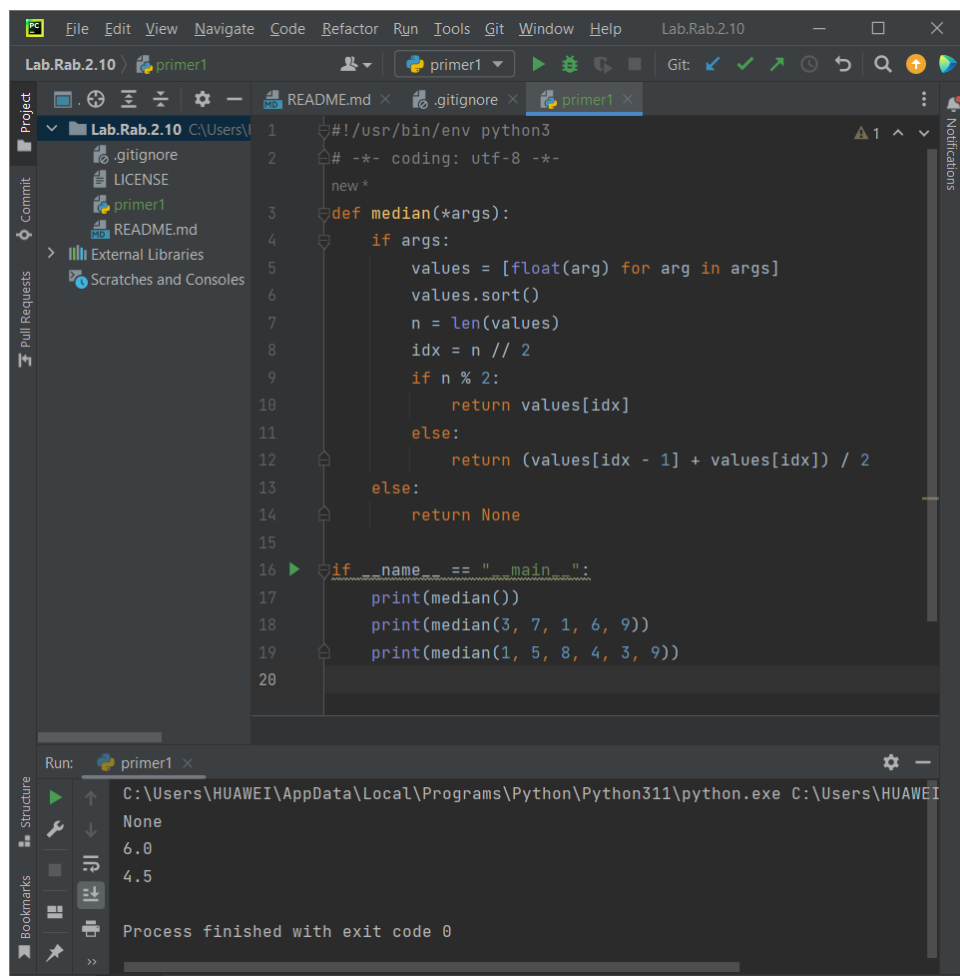
Создал проект PyCharm в папке репозитория.

Работа с примером №1.

Добавил новый файл *primer1.py*

Условие примера: Разработать функцию для определения медианы значений аргументов функции. Если функции передается пустой список аргументов, то она должна возвращать значение None .

Медианой (серединой) набора чисел называется число стоящее посередине упорядоченного по возрастанию ряда чисел. Если количество чисел в ряду чётное, то медианой ряда является полусумма двух стоящих посередине чисел. Применяется в математической статистике — число, характеризующее выборку (например, набор чисел), также используется для вычисления медианной зарплаты.



The screenshot shows the PyCharm IDE interface. The main editor displays the `primer1.py` file with the following code:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 new *
4 def median(*args):
5     if args:
6         values = [float(arg) for arg in args]
7         values.sort()
8         n = len(values)
9         idx = n // 2
10        if n % 2:
11            return values[idx]
12        else:
13            return (values[idx - 1] + values[idx]) / 2
14    else:
15        return None
16
17 if __name__ == "__main__":
18     print(median())
19     print(median(3, 7, 1, 6, 9))
20     print(median(1, 5, 8, 4, 3, 9))
```

The bottom panel shows the output of the program:

```
Run: primer1 x
C:\Users\HUAWEI\AppData\Local\Programs\Python\Python311\python.exe C:\Users\HUAWEI
None
6.0
4.5
Process finished with exit code 0
```

Рисунок 3 – Программа и ее результат

Зафиксировал данные изменения.

Задание 4.

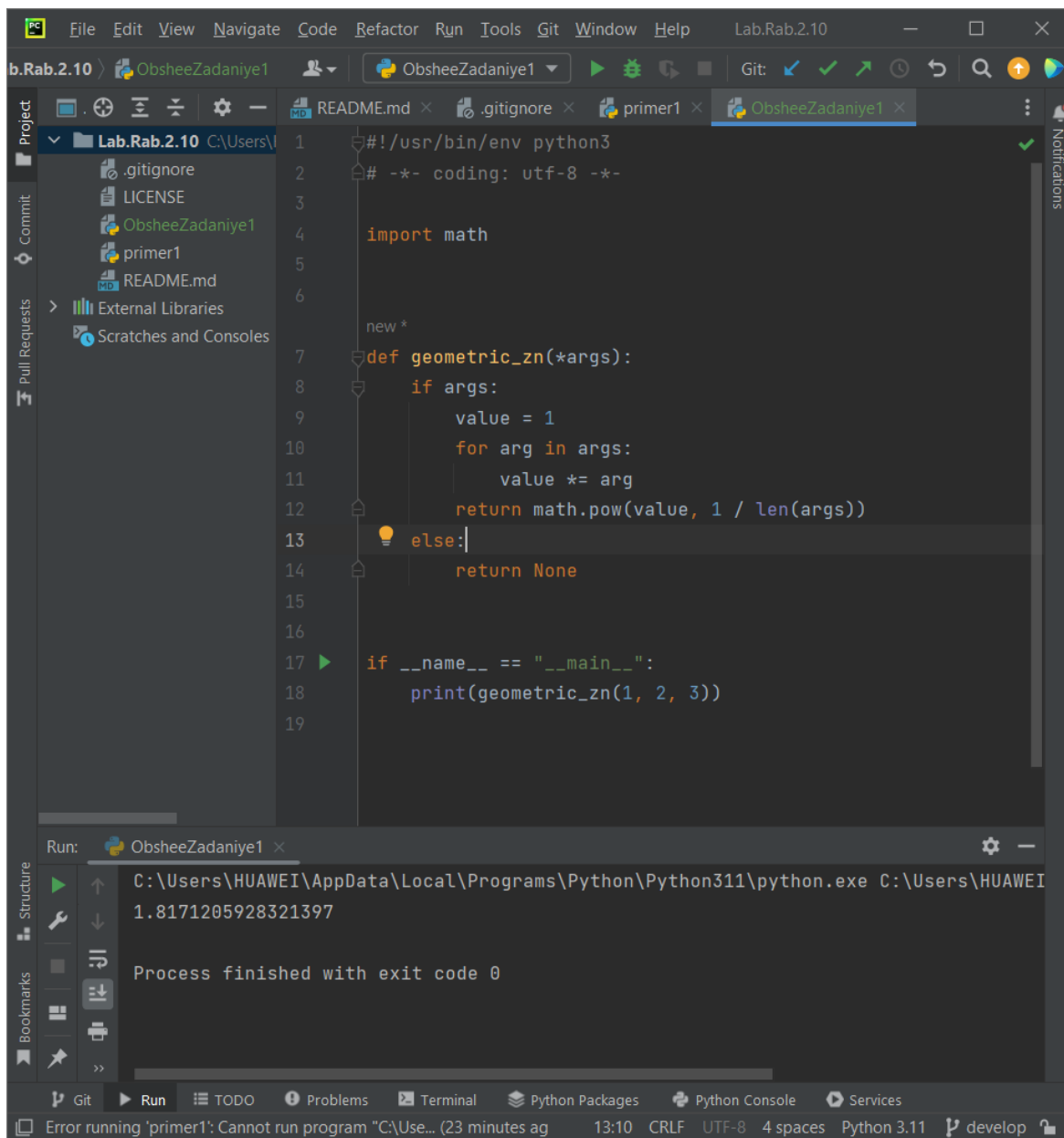
Выполнение общего задания.

Создал новый файл под названием *ObsheeZanadiye1.py*

Условие общего задания:

Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов a_1, a_2, \dots, a_n

$$G = \sqrt[n]{\prod_{k=1}^n a_k}. \quad (1)$$



The screenshot shows a code editor with a Python file named `ObsheeZadaniye1.py`. The code defines a function `geometric_zn(*args)` that calculates the geometric mean of its arguments. It uses a loop to multiply all arguments into a single value, then raises that value to the power of $1/n$ where n is the number of arguments. A main block at the bottom prints the result of `geometric_zn(1, 2, 3)`.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import math
5
6
7  def geometric_zn(*args):
8      if args:
9          value = 1
10         for arg in args:
11             value *= arg
12         return math.pow(value, 1 / len(args))
13     else:
14         return None
15
16
17 if __name__ == "__main__":
18     print(geometric_zn(1, 2, 3))
19
```

The Run console at the bottom shows the command executed: `C:\Users\HUAWEI\AppData\Local\Programs\Python\Python311\python.exe C:\Users\HUAWEI\...` and the output: `1.8171205928321397`. It also indicates that the process finished with exit code 0.

Рисунок 4 – Программа и выполнение

Зафиксировал данные изменения.

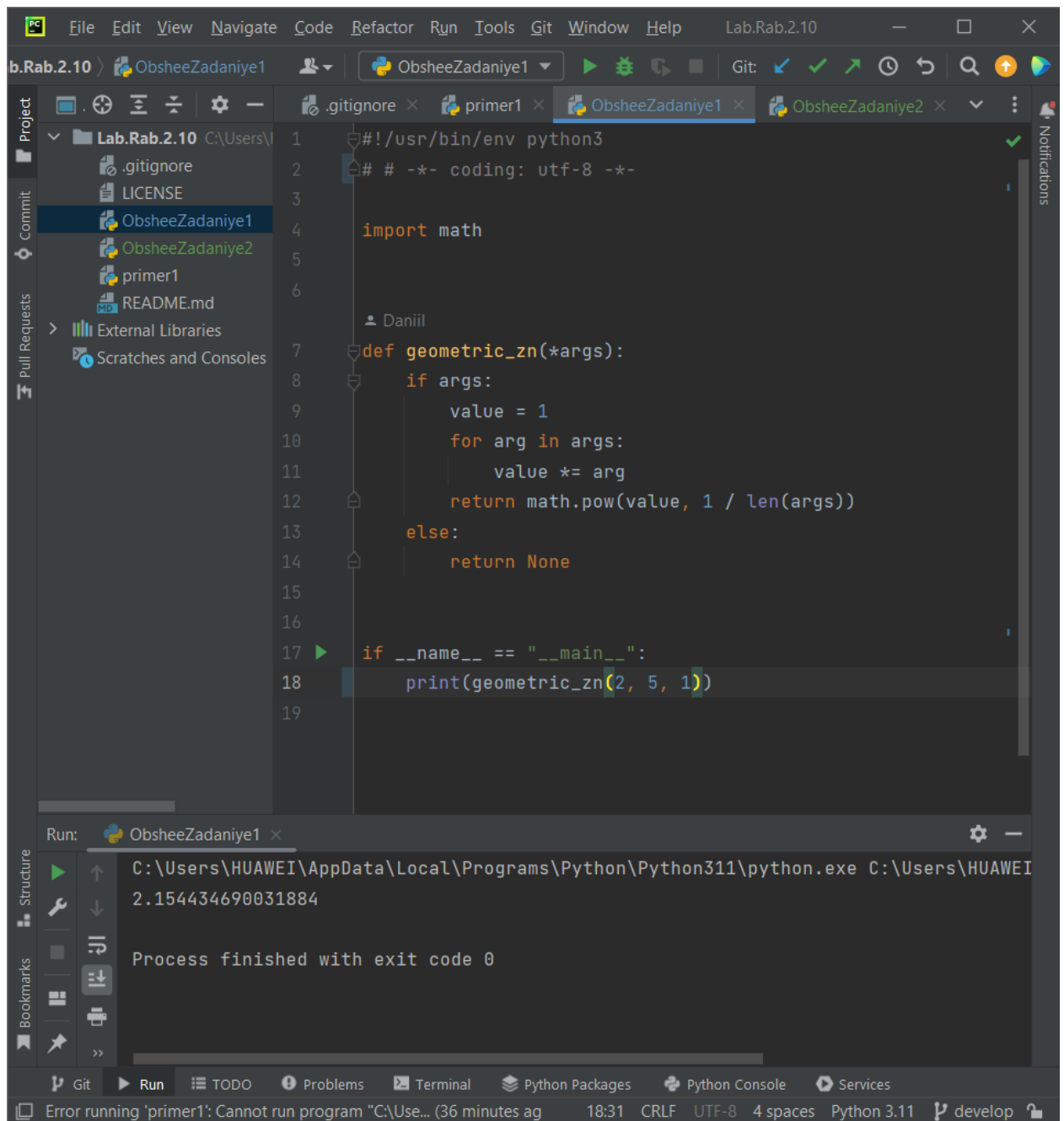


Рисунок 5 – Программы и выполнение (при других данных)

Задание 5.

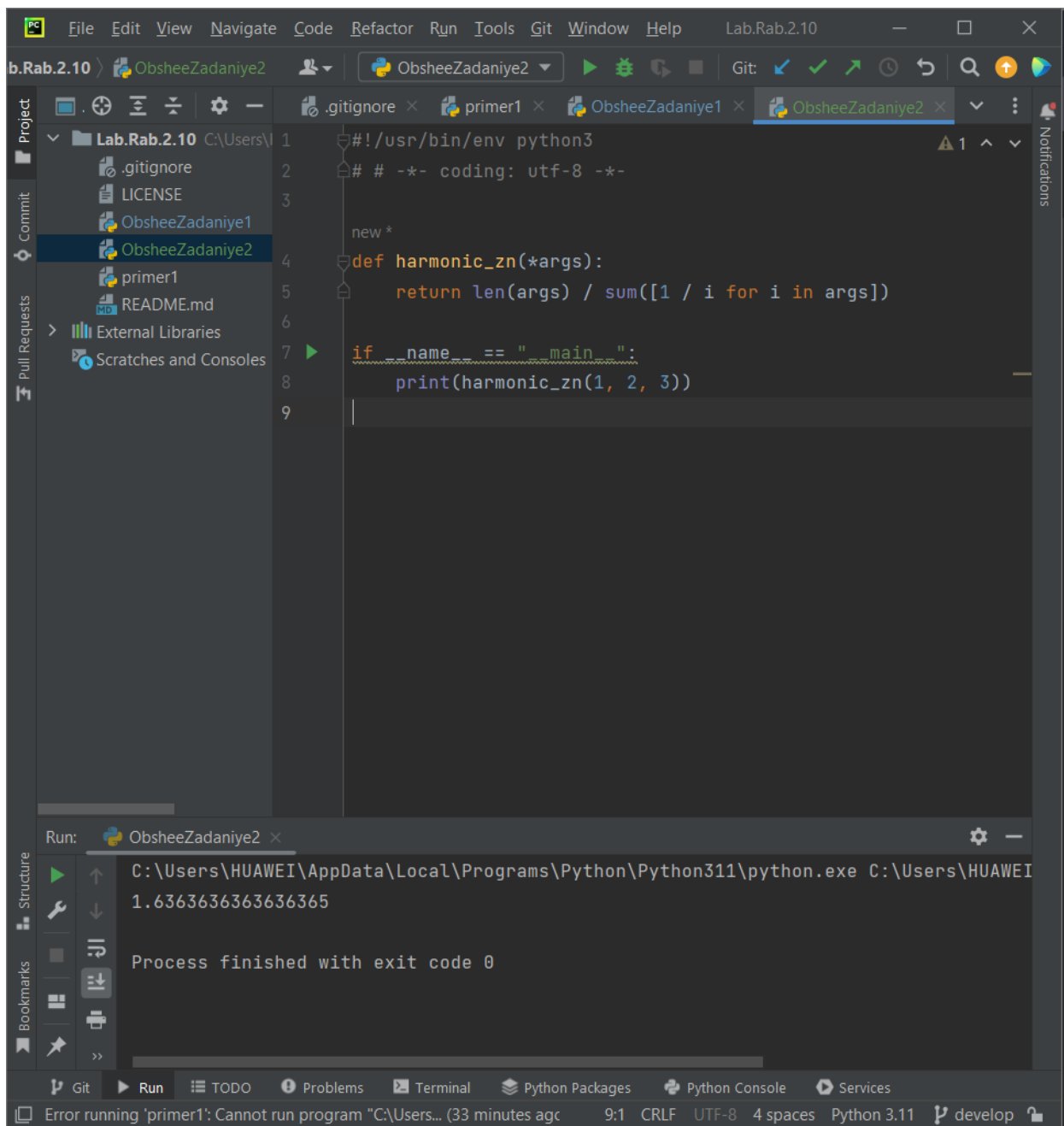
Выполнение общего задания.

Создал новый файл под названием *ObsheeZanadiye2.py*

Условие общего задания:

Решить поставленную задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов a_1, a_2, \dots, a_n

$$\frac{n}{H} = \sum_{k=1}^n \frac{1}{a_k}. \quad (2)$$



The screenshot shows an IDE window with a Python script and its execution output. The script defines a function `harmonic_zn` that calculates the harmonic mean of its arguments. The output shows the function being called with arguments 1, 2, and 3, resulting in the value 1.6363636363636365.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 new *
5 def harmonic_zn(*args):
6     return len(args) / sum([1 / i for i in args])
7
8 if __name__ == "__main__":
9     print(harmonic_zn(1, 2, 3))
```

Run: ObsheeZadaniye2

```
C:\Users\HUAWEI\AppData\Local\Programs\Python\Python311\python.exe C:\Users\HUAWEI
1.6363636363636365

Process finished with exit code 0
```

Рисунок 6 – Программа и выполнение

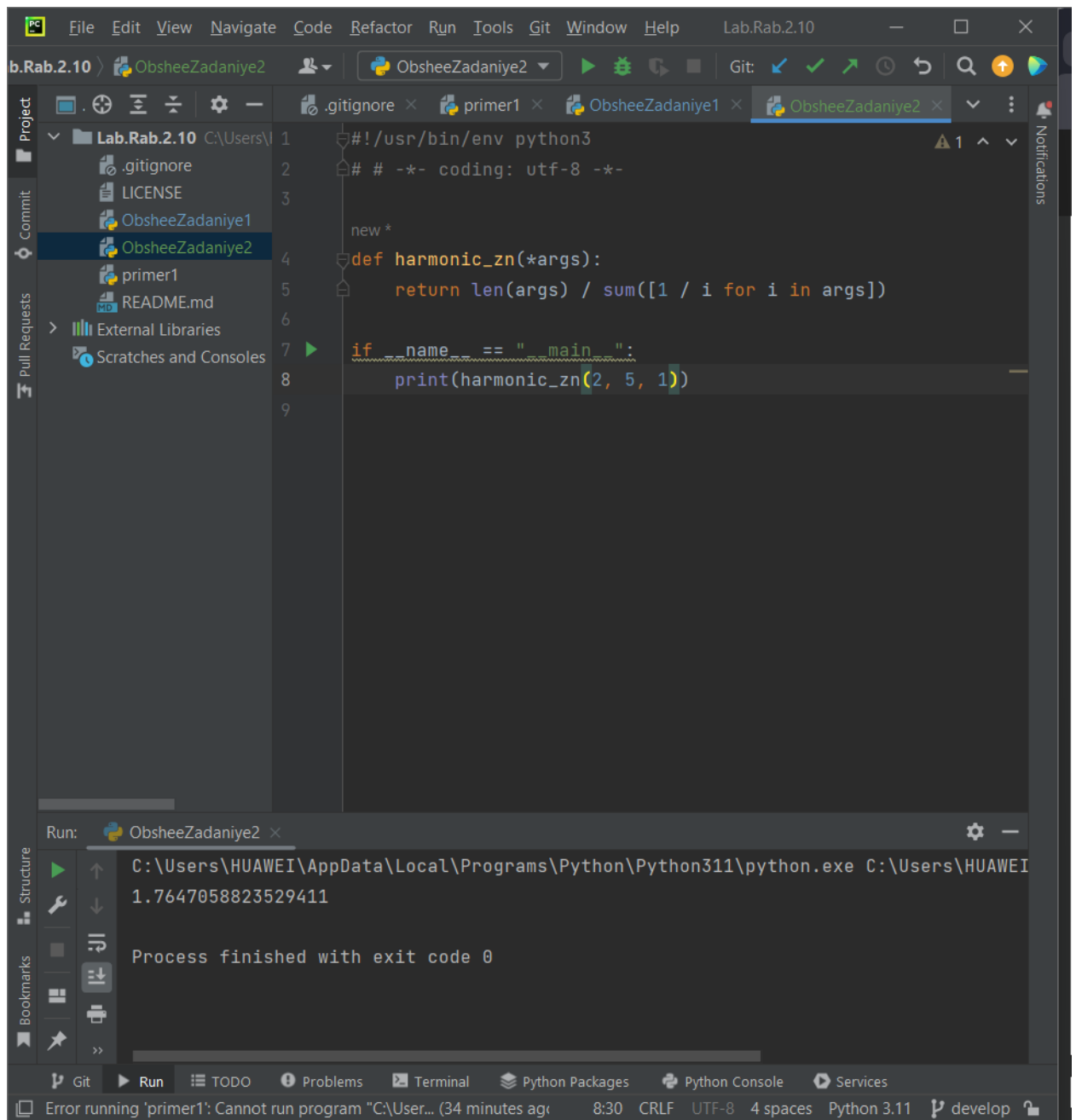


Рисунок 7 – Программы и выполнение (при других данных)

Зафиксировал данные изменения.

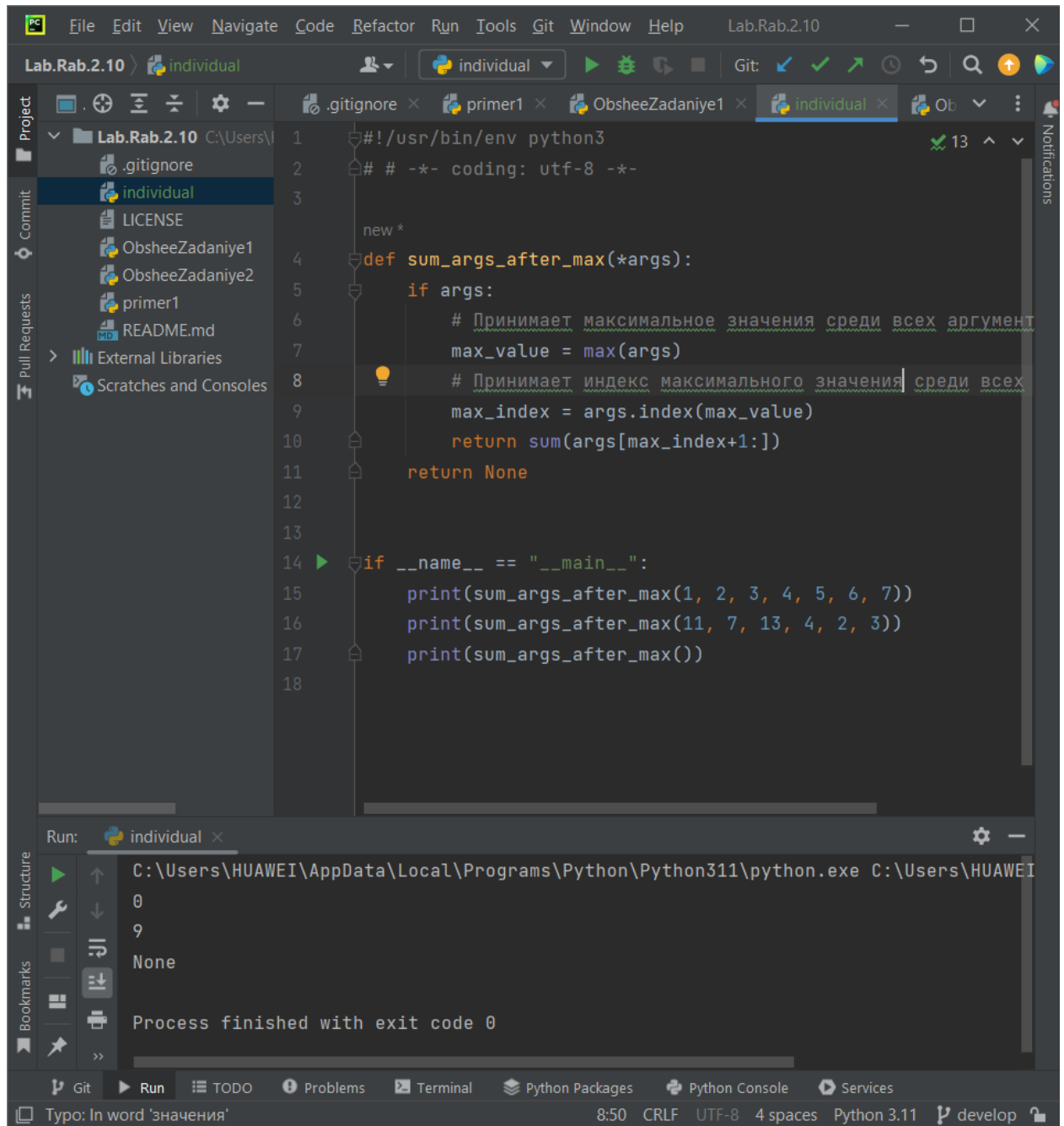
Задание 6.

Выполнение индивидуального задания.

Создал новый файл под названием *individual.py*

Вариант 12 (по списку группы).

Условие общего задания: найти сумму аргументов, расположенных после максимального аргумента.



The screenshot shows a code editor with a Python script named `individual.py`. The script defines a function `sum_args_after_max(*args)` that calculates the sum of arguments following the maximum value. The `__main__` block tests the function with three sets of arguments: `(1, 2, 3, 4, 5, 6, 7)`, `(11, 7, 13, 4, 2, 3)`, and an empty tuple `()`.

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

def sum_args_after_max(*args):
    if args:
        # Принимает максимальное значения среди всех аргументов
        max_value = max(args)
        # Принимает индекс максимального значения среди всех
        max_index = args.index(max_value)
        return sum(args[max_index+1:])
    return None

if __name__ == "__main__":
    print(sum_args_after_max(1, 2, 3, 4, 5, 6, 7))
    print(sum_args_after_max(11, 7, 13, 4, 2, 3))
    print(sum_args_after_max())
```

The Run window at the bottom shows the execution output:

```
C:\Users\HUAWEI\AppData\Local\Programs\Python\Python311\python.exe C:\Users\HUAWEI
0
9
None
Process finished with exit code 0
```

Рисунок 8 – Программа и выполнение

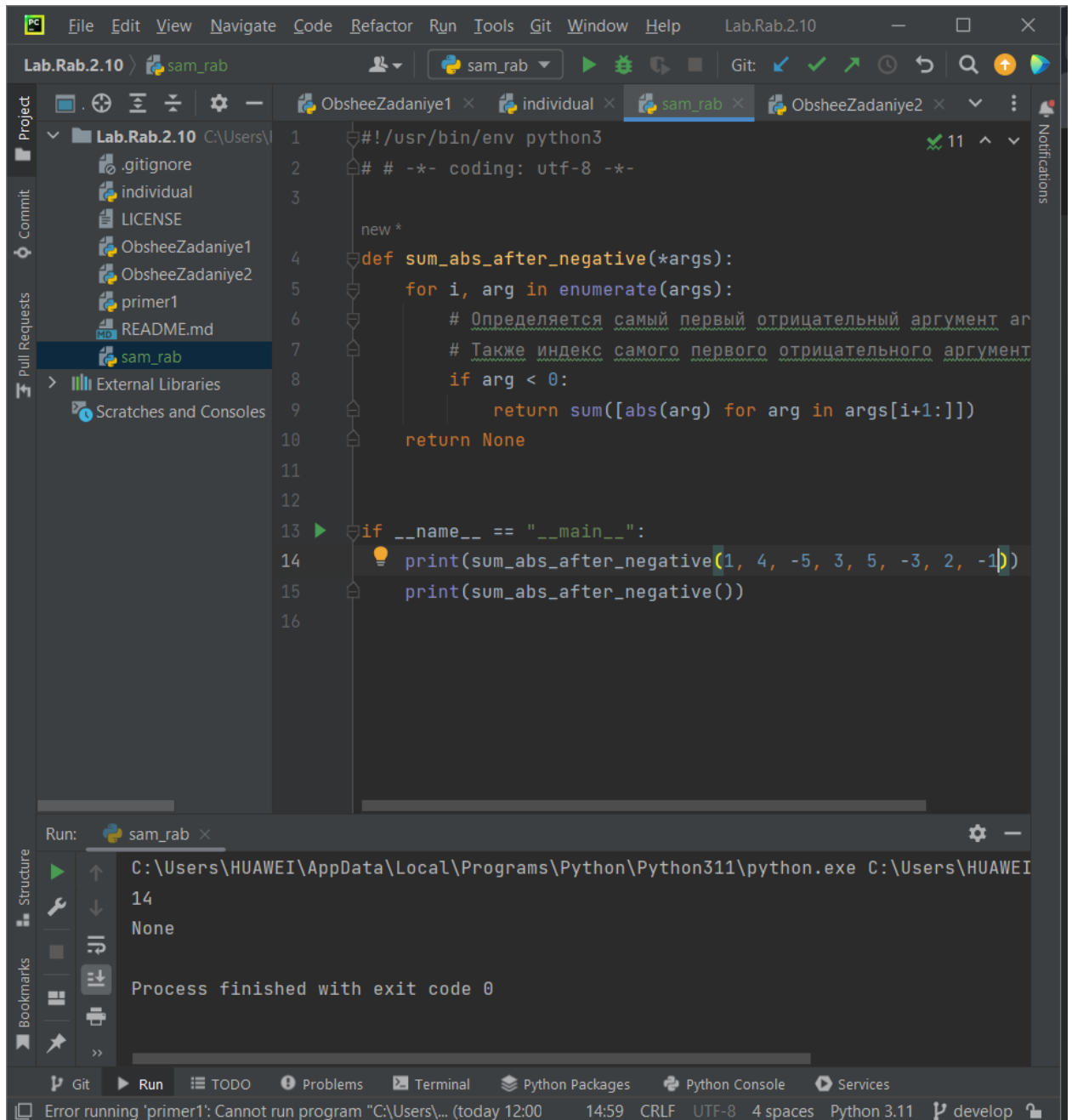
Зафиксировал данные изменения.

Задание 7.

Самостоятельно подберите или придумайте задачу с переменным числом именованных аргументов. Приведите решение этой задачи.

Создал новый файл под названием *sam_rab.py*

Условие задачи: найти сумму модулей аргументов, расположенных после первого отрицательного аргумента.



The screenshot shows a code editor with a Python script named *sam_rab.py*. The script defines a function *sum_abs_after_negative* that takes an arbitrary number of arguments. It iterates through the arguments to find the first negative one. Once found, it returns the sum of the absolute values of all subsequent arguments. If no negative argument is found, it returns *None*. The script also includes a main block that tests the function with the arguments *(1, 4, -5, 3, 5, -3, 2, -1)*.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  new *
5  def sum_abs_after_negative(*args):
6      for i, arg in enumerate(args):
7          # Определяется самый первый отрицательный аргумент arg
8          # Также индекс самого первого отрицательного аргумент
9          if arg < 0:
10             return sum([abs(arg) for arg in args[i+1:]])
11         return None
12
13 if __name__ == "__main__":
14     print(sum_abs_after_negative(1, 4, -5, 3, 5, -3, 2, -1))
15     print(sum_abs_after_negative())
16
```

The output of the script is shown in the Run console:

```
Run: sam_rab x
C:\Users\HUAWEI\AppData\Local\Programs\Python\Python311\python.exe C:\Users\HUAWEI
14
None
Process finished with exit code 0
```

Рисунок 9 – Программа и выполнение

Зафиксировал данные изменения.

Задание 8.

Слил ветку develop с веткой main и отправил на удаленный сервер.

```
C:\Users\HUAWEI\Lab.Rab.2.10>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

C:\Users\HUAWEI\Lab.Rab.2.10>git merge develop
Merge made by the 'ort' strategy.
 ObsheeZadaniye1 | 18 ++++++
 ObsheeZadaniye2 |  8 ++++++
 individual       | 17 ++++++
 primer1         | 19 ++++++
 sam_rab         | 15 ++++++
 5 files changed, 77 insertions(+)
 create mode 100644 ObsheeZadaniye1
 create mode 100644 ObsheeZadaniye2
 create mode 100644 individual
 create mode 100644 primer1
 create mode 100644 sam_rab
```

Рисунок 10 – Слияние веток

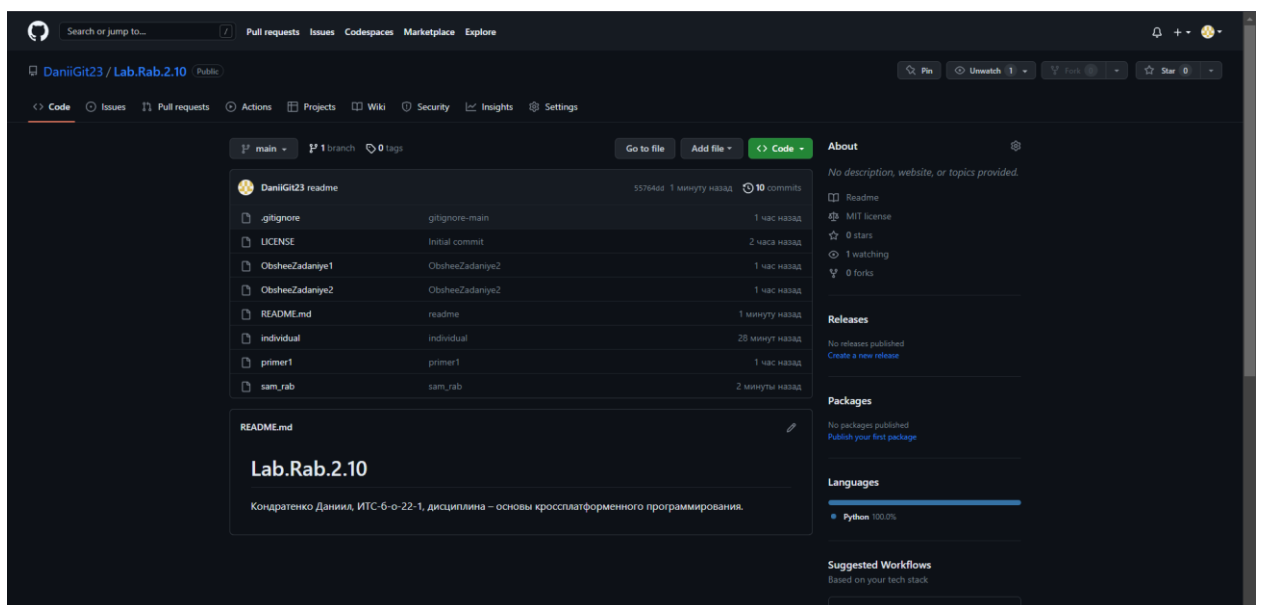


Рисунок 11 – Удаленный сервер

Ссылка на репозиторий: <https://github.com/DaniiGit23/Lab.Rab.2.10.git>

Ответы на контрольные вопросы:

1) Какие аргументы называются позиционными в Python?

Позиционные аргументы в Python - это тип аргументов функции, когда переданные значения связываются с параметрами функции в порядке их

передачи. То есть первый аргумент связывается с первым параметром, второй аргумент со вторым параметром и т.д. Последовательность передачи аргументов важна и должна точно соответствовать порядку параметров функции.

2) Какие аргументы называются именованными в Python?

Именованные аргументы в Python - это аргументы, которые передаются в функцию вместе с их именем или ключом, и которые позволяют явно указать, какой аргумент используется для данного параметра. Они обычно указываются в виде имя_аргумента=значение и должны следовать после всех позиционных аргументов.

3) Для чего используется оператор *?

Оператор * используется в Python для распаковки последовательностей (например, списков или кортежей) и передачи их элементов в функцию в качестве отдельных аргументов. Он также может использоваться для распаковки аргументов внутри функции или метода.

4) Каково назначение конструкций *args и **kwargs ?

*args и **kwargs предназначены для передачи произвольного количества аргументов в функцию.

Вывод: приобретение навыков по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.