

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 ЛАБОРАТОРНОЙ РАБОТЕ №2.10
дисциплины «Основы кроссплатформенного программирования»

Выполнила:
Мурашко Анастасия Юрьевна
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., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: функции с переменным числом параметров в Python.

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

Ход работы:

Задание 1. Создала проект PyCharm в папке репозитория. Приступила к работе с примером. Добавил новый файл primer.py.

```
C:\Users\user>git clone https://github.com/welli/laba2-10.git
Cloning into 'laba2-10'...
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\user>cd C:\Users\user\laba2-10

C:\Users\user\laba2-10>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/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/user/laba2-10/.git/hooks]
```

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

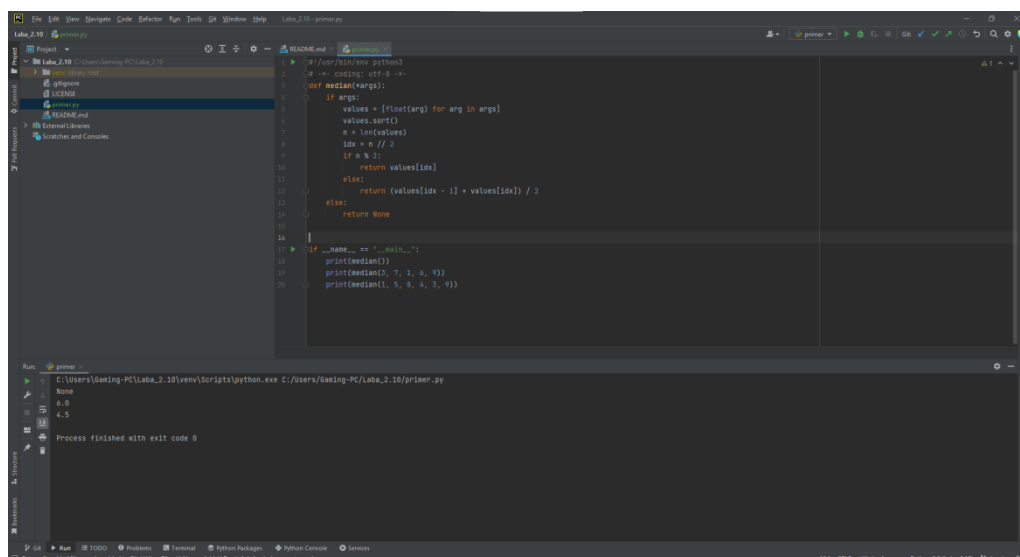


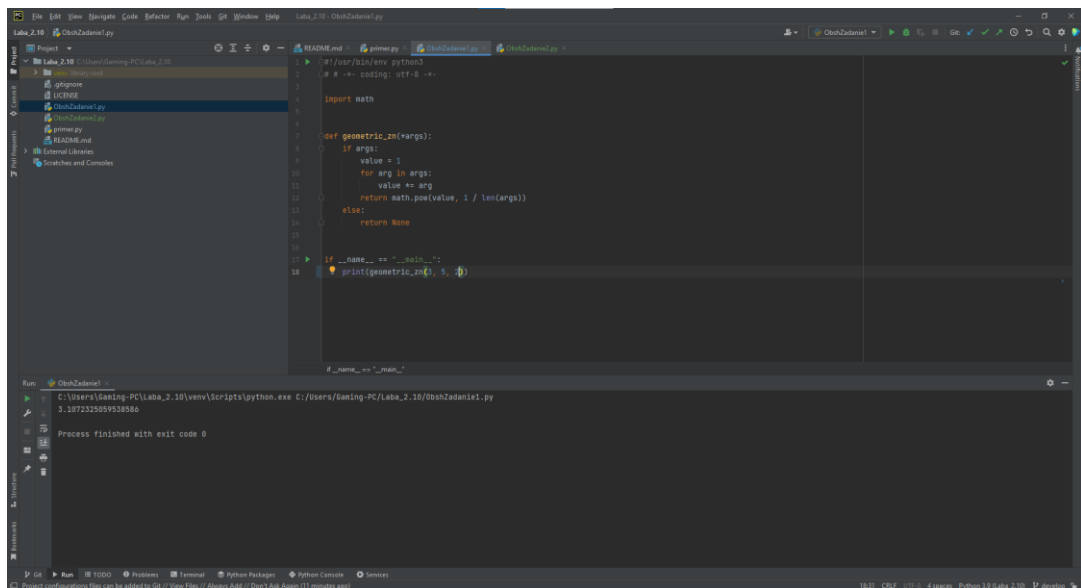
Рисунок 3. Реализация примера

Задание 4. Создала новый файл под названием ObshZadanie1. Приступила к работе с общим заданием №1.

Условие примера:

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

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

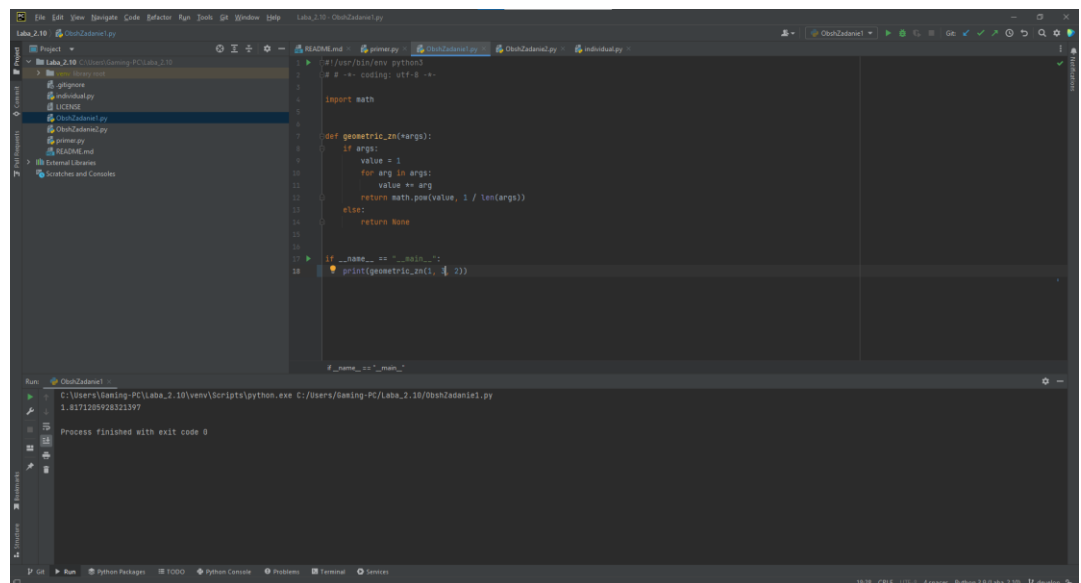


The screenshot shows a code editor with a file named 'ObshZadanie1.py'. The code defines a function 'geometric_mean' that calculates the geometric mean of its arguments. It uses a loop to calculate the product of all arguments and then returns the nth root of the product. The function is tested with a single example: 'print(geometric_mean(5, 5))'. The output in the console is '3.1622776601683795'.

```
def geometric_mean(*args):
    if args:
        value = 1
        for arg in args:
            value *= arg
        return math.pow(value, 1 / len(args))
    else:
        return None

if __name__ == "__main__":
    print(geometric_mean(5, 5))
```

Рисунок 4. Реализация общего задания 1



The screenshot shows the same code editor with 'ObshZadanie1.py'. The code is identical to the previous one, but the test case is changed to 'print(geometric_mean(1, 2))'. The output in the console is '1.4142135623730951'.

```
def geometric_mean(*args):
    if args:
        value = 1
        for arg in args:
            value *= arg
        return math.pow(value, 1 / len(args))
    else:
        return None

if __name__ == "__main__":
    print(geometric_mean(1, 2))
```

Рисунок 5. Реализация общего задания 1 с другими данными

Задание 5. Создала новый файл под названием ObshZadanie2. Приступил к работе с общим заданием №2.

Условие примера:

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

$$\frac{n}{H} = \sum_{k=1}^n \frac{1}{a_k}.$$

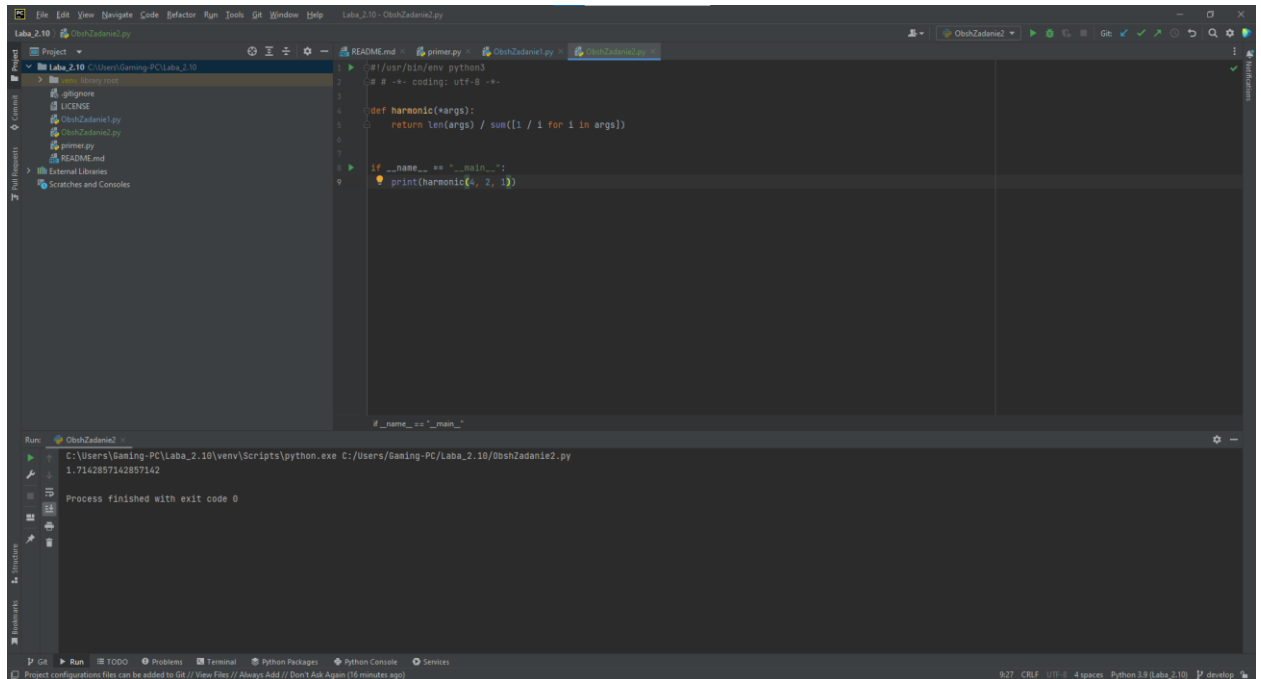


Рисунок 6. Реализация общего задания 2

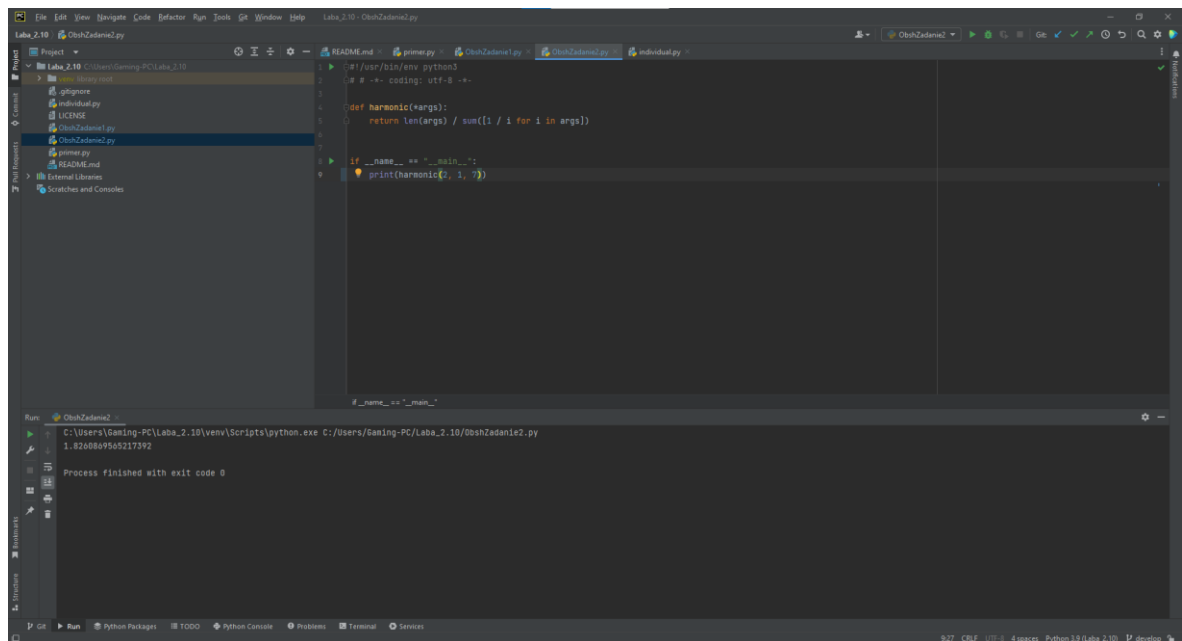


Рисунок 7. Реализация общего задания 2 с другими данными

Задание 6.

Индивидуальное задание

Вариант 10

Создала новый файл под названием individual.py.

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

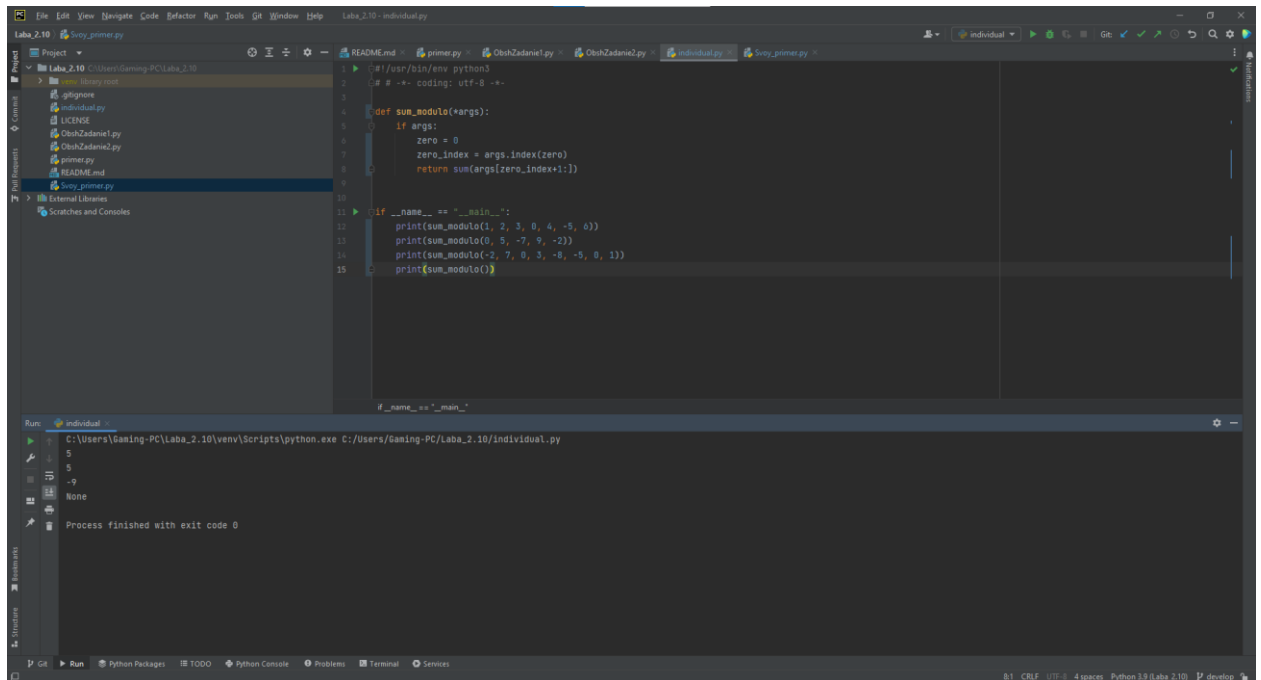


Рисунок 8. Программа индивидуального задания

Задание 7. Создала новый файл под названием `Svoy_primer`. Выбрала пример задания из списка индивидуальных заданий.

Условие примера: Сумму аргументов, расположенных после минимального аргумента.

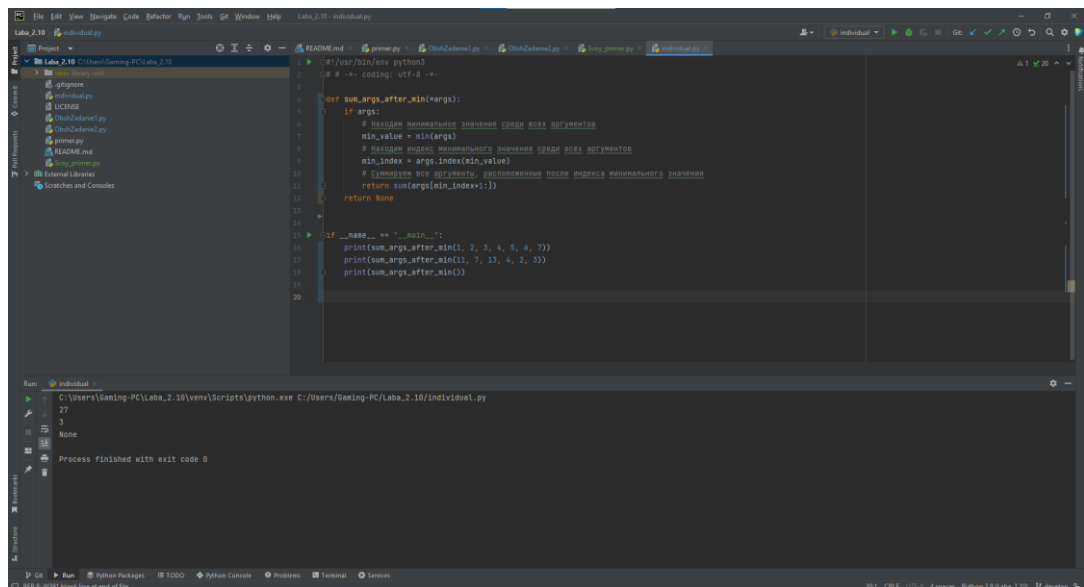


Рисунок 9. Программа выбранного задания

Задание 7.

После выполнения работы на ветке develop, слила ее с веткой main и отправила изменения на удаленный сервер.

Ответы на контрольные вопросы:

1. Какие аргументы называются позиционными в Python?

Позиционные аргументы - это аргументы функции, передаваемые в порядке их указания при вызове функции. Значение каждого позиционного аргумента соответствует порядку его указания в списке аргументов функции. Последовательность передачи аргументов важна и должна точно соответствовать порядку параметров функции.

2. Какие аргументы называются именованными в Python?

Именованные аргументы - это аргументы функции, которые передаются в форме name=value. В отличие от позиционных аргументов, порядок их передачи не важен, поскольку они связываются с именами параметров функции, а не с их позицией.

3. Для чего используется оператор *?

Оператор * используется в Python для распаковки последовательностей (например, списка или кортежа) или коллекций

(например, словаря) в отдельные элементы. Он также может использоваться в определении функций для передачи переменного числа аргументов (позиционных или именованных).

4. Каково назначение конструкций `*args` и `**kwargs` ?

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

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