

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.10
дисциплины
«Программирование на Python»

Выполнил:
Костукайло Кирилл Николаевич
2 курс, группа ИВТ-б-о-21-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Проверил:
Кафедры инфокоммуникаций, старший
преподаватель
Воронкин Р.А.

(подпись)

Отчёт защищён с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

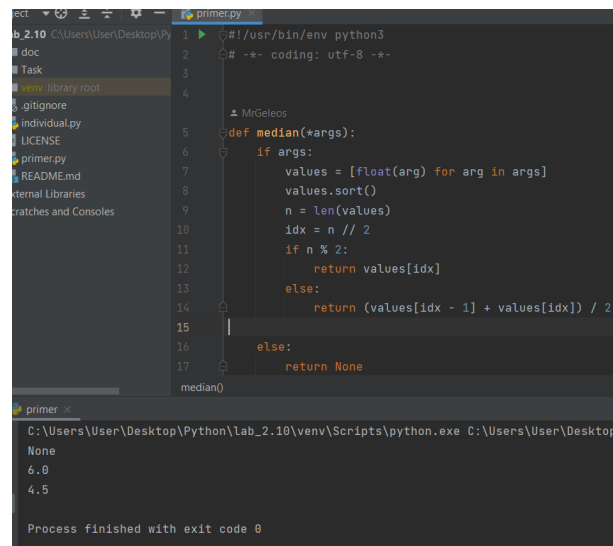
Тема: **Функции с переменным числом параметров в Python**

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

Ход работы:

1. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python, клонировал созданного репозитория.

2. Создайте проект PyCharm в папке репозитория и проработал пример лабораторной работы.



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

Output in console:

```
None
6.0
4.5
Process finished with exit code 0
```

Рисунок 1. Пример

Выполнил задачи:

Задача 1:

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

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

Если функции передается пустой список аргументов, то она должна возвращать значение `None`.

Рисунок 2.1. Задание

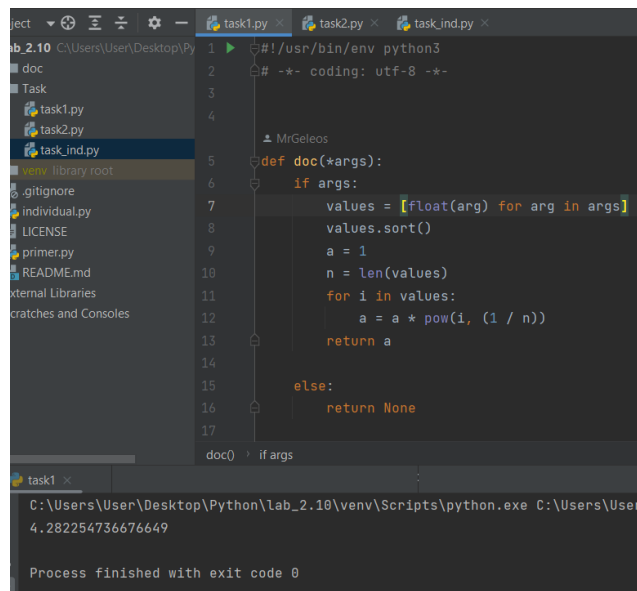


Рисунок 2.2. Задача 1

Задача 2:

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

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

Если функции передается пустой список аргументов, то она должна возвращать значение `None`.

Рисунок 3.1. Задание

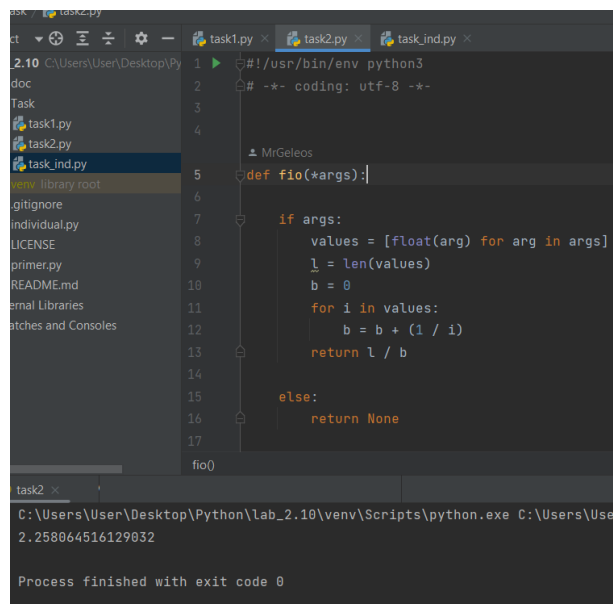


Рисунок 3.2. Задача 2

Задача 3: самостоятельно подберите или придумайте задачу с переменным числом именованных аргументов. Приведите решение этой задачи.

Контрольные вопросы:

1. Какие аргументы называются позиционными в Python?

Аргументы, которые передаются без указания имен называются позиционными, потому что именно по позиции, расположению аргумента, функция понимает, какому параметру он соответствует.

2. Какие аргументы называются именованными в Python?

Аргументы, передаваемые с именами, называются именованными. При вызове функции можно использовать имена параметров из ее определения.

3. Для чего используется оператор *?

Оператор * чаще всего ассоциируется у людей с операцией умножения, но в Python он имеет и другой смысл. Этот оператор позволяет «распаковывать» объекты, внутри которых хранятся некие элементы. Вот пример:

```
a = [1, 2, 3]
```

```
b = [*a, 4, 5, 6]
```

```
print(b) # [1, 2, 3, 4, 5, 6]
```

Тут берётся содержимое списка a, распаковывается, и помещается в список b.

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

Итак, мы знаем о том, что оператор «звёздочка» в Python способен «вытаскивать» из объектов составляющие их элементы. Знаем мы и о том, что существует два вида параметров функций. А именно, *args — это сокращение от «arguments» (аргументы), а **kwargs — сокращение от «keyword arguments» (именованные аргументы).

Каждая из этих конструкций используется для распаковки аргументов соответствующего типа, позволяя вызывать функции со списком аргументов переменной длины. Важно помнить, что «args» — это всего лишь набор символов, которым принято обозначать аргументы. Самое главное тут — это оператор *. А то, что именно идёт после него, особой роли не играет. Благодаря использованию * мы создали список позиционных аргументов на

основе того, что было передано функции при вызове. После того, как мы разобрались с `*args`, с пониманием `**kwargs` проблем быть уже не должно. Имя, опять же, значения не имеет. Главное — это два символа `**`. Благодаря им создаётся словарь, в котором содержатся именованные аргументы, переданные функции при её вызове.

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