

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт цифрового развития**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.10

Функции с переменным числом параметров в Python

Выполнил студент группы

ИВТ-б-о-21-1 (2)

Стригалов Д.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

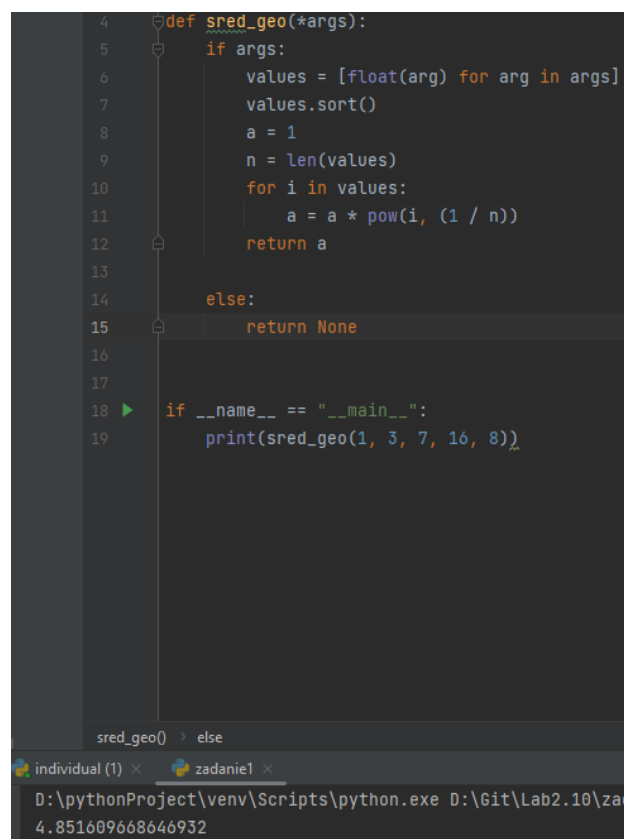
Ставрополь 2022

Ход работы:

Задание №1. Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов $a_1, a_2 \dots a_n$. Если функции передается пустой список аргументов, то она должна возвращать значение None.

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

Рисунок 1 – Формула для задания №1



```
4 def sred_geo(*args):
5     if args:
6         values = [float(arg) for arg in args]
7         values.sort()
8         a = 1
9         n = len(values)
10        for i in values:
11            a = a * pow(i, (1 / n))
12        return a
13    else:
14        return None
15
16
17
18 if __name__ == "__main__":
19     print(sred_geo(1, 3, 7, 16, 8))
```

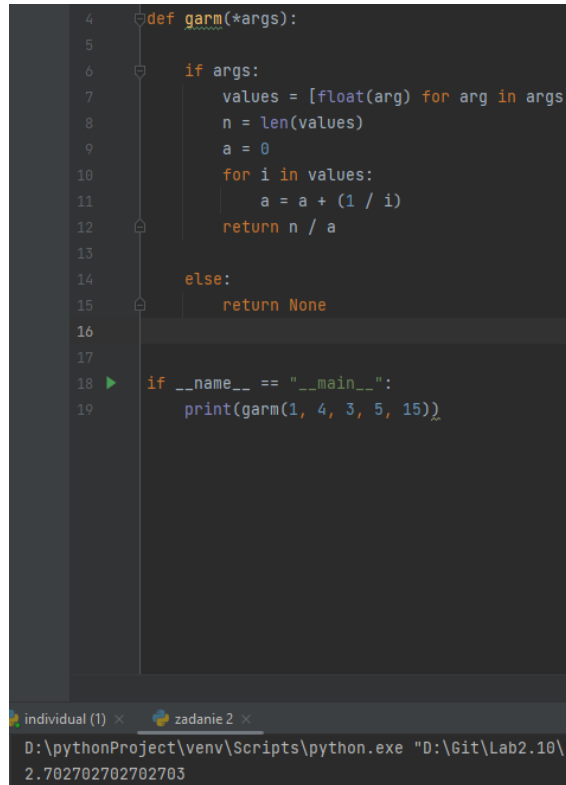
The screenshot shows a Python IDE with a dark theme. The code defines a function `sred_geo` that takes an arbitrary number of arguments. If there are arguments, it converts them to floats, sorts them, and calculates the geometric mean by multiplying each value to the power of $1/n$. If no arguments are provided, it returns `None`. Below the function definition, a main block calls `sred_geo(1, 3, 7, 16, 8)`. At the bottom, a terminal window shows the command `D:\pythonProject\venv\Scripts\python.exe D:\Git\Lab2.10\za...` and the output `4.851609668646932`.

Рисунок 2 – Результат работы программы

Задание №2. Решить поставленную задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов $a_1, a_2 \dots a_n$. Если функции передается пустой список аргументов, то она должна возвращать значение None.

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

Рисунок 3 – Формула для задания №2



```

4  def garm(*args):
5
6      if args:
7          values = [float(arg) for arg in args]
8          n = len(values)
9          a = 0
10         for i in values:
11             a = a + (1 / i)
12         return n / a
13
14     else:
15         return None
16
17
18  if __name__ == "__main__":
19      print(garm(1, 4, 3, 5, 15))

```

individual (1) × zadanie 2 ×
D:\pythonProject\venv\Scripts\python.exe "D:\Git\Lab2.10\z
2.702702702702703

Рисунок 4 – Результат работы программы

Индивидуальное задание. Напишите функцию, принимающую произвольное количество аргументов, и возвращающую требуемое значение. Если функции передается пустой список аргументов, то она должна возвращать значение None. В процессе решения не использовать преобразования конструкции *args в список или иную структуру данных.

Задание: 1. Произведение аргументов, расположенных между максимальным и минимальным аргументами.

```
4 def multiplic(*args):
5     amin = 0
6     amax = 0
7     maxim = 0
8     proiz = 1
9     if args:
10        for i in args:
11            if i > maxim:
12                maxim = i
13                amax = args.index(i)
14
15        mini = args[0]
16        for i in args:
17            if i < mini:
18                mini = i
19                amin = args.index(i)
20
21        for i in args[amin+1:amax]:
22            proiz *= i
23        return 'Произведение =', proiz
24
25     else:
26        return None
27
28
29 if __name__ == "__main__":
30     print(multiplic(10, 150, 5, 400, 13, 555))
```

individual (1) x individual x
D:\pythonProject\venv\Scripts\python.exe D:\Git\Lab2.10\individual.py
('Произведение =', 5200)

Рисунок 5 – Результат выполнения программы

Вывод: в результате выполнения лабораторной работы были приобретены практические навыки и теоретические сведения по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.

Ответы на 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.