

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.10

Функции с переменным числом параметров в Python

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2021

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

Ход работы: Пример 1

<https://github.com/Mirror-Shard/L2.10>

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

1. Передаваемые в функцию значения:

```
print(median())  
print(median(3, 7, 1, 6, 9))  
print(median(1, 5, 8, 4, 3, 9))
```

Рисунок 1 – Пример 1, значения

2. Функция успешно определяет медиану значений

```
None  
6.0  
4.5
```

Рисунок 2 – Результат работы примера

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.

1. Создал функцию, вычисляющую среднее геометрическое, программа выводит:

```
Введите числа в массив через пробел:  
1 2 3 4 5 6 7 8 9 10  
4.528728688116765
```

Рисунок 4 – Среднее геометрическое

2. Программа выводит 0, если в списке есть 0:

```
Введите числа в массив через пробел:  
1 2 3 4 0  
0.0
```

Рисунок 5 – Среднее геометрическое с нулём

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.

1. Функция успешно выводит среднее гармоническое для простого перечисления:

```
Введите числа в массив через пробел (нули запрещены):  
1 2 3 4 5  
2.18978102189781
```

Рисунок 6 – Среднее гармоническое

2. Нули в списке запрещены, поэтому при наличии нуля – программа выводит соответствующее сообщение:

```
Введите числа в массив через пробел (нули запрещены):  
1 2 0 4 5  
НУЛИ ЗАПРЕЩЕНЫ!!!
```

Рисунок 7 – Массив с нестандартными числами

Индивидуальное задание

Напишите функцию, принимающую произвольное количество аргументов, и возвращающую сумму аргументов, расположенных между первым и последним нулевыми аргументами.

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.

1. Программа успешно считает сумму между первым и вторым нулём, остальные игнорируются:

```
Введите числа в массив через пробел, в массиве должно быть минимум два нуля:  
1 2 3 4 0 5 6 0 7  
11.0
```

Рисунок 8 – Сумма элементов между нулями

2. Если в программе 1 ноль, то выводит сумму элементов после него:

```
Введите числа в массив через пробел, в массиве должно быть минимум два нуля:  
1 2 3 0 5 6  
11.0
```

Рисунок 9 – Сумма элементов после нуля

Контрольные вопросы:

1. Какие аргументы называются позиционными в Python? Позиционные аргументы обрабатываются слева направо. То есть

оказывается, что позиция аргумента, переданного функции, находится в прямом соответствии с позицией параметра, использованного в заголовке функции при её объявлении.

2. Какие аргументы называются именованными в Python? Именованные аргументы передают функциям с указанием имён этих аргументов, соответствующих тем именам, которые им назначены при объявлении функции.

3. Для чего используется оператор `*?` Этот оператор позволяет «распаковывать» объекты, внутри которых хранятся некие элементы.

4. Каково назначение конструкций `*args` и `**kwargs`? При применении конструкции `*args` в параметр `args` попадают позиционные аргументы, представляемые в виде кортежа. При применении `**kwargs` в `kwargs` попадают именованные аргументы, представленные в виде словаря.

Вывод: в ходе работы приобрёл навыки по работе с функциями с переменным числом аргументов при написании программ с помощью языка программирования Python версии 3.