

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

Кафедра инфокоммуникаций

**Отчет по лабораторной работе №5
Функции с переменным числом параметров в Python.**

по дисциплине «Технологии программирования и алгоритмизации»

Выполнила студентка группы ИВТ-б-о-20-1

Новикова В.С. « » _____ 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 3х.

Ход работы:

Ссылка на репозиторий: https://github.com/Valentina1502/LABA_5

Пример 1. (рис. 1).

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

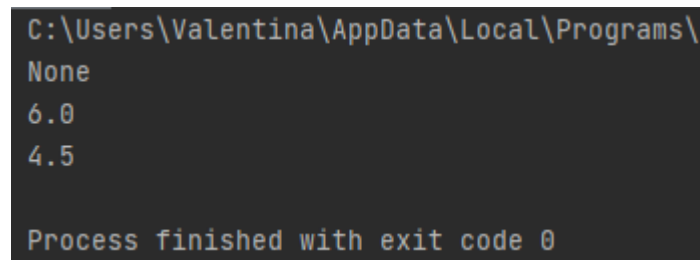
Код:

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

def median(*args):
    if args:
        values = [float(arg) for arg in args]
        values.sort()

        n = len(values)
        idx = n // 2
        if n % 2:
            return values[idx]
        else:
            return (values[idx - 1] + values[idx]) / 2
    else:
        return None

if __name__ == "__main__":
    print(median())
    print(median(3, 7, 1, 6, 9))
    print(median(1, 5, 8, 4, 3, 9))
```



```
C:\Users\Valentina\AppData\Local\Programs\Python\Python39\python.exe
None
6.0
4.5

Process finished with exit code 0
```

Рисунок 1 – Пример 1

Задание 1 (рис. 2-3):

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 2 – Средне геометрическое

Код:

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

def G(*args):
    if args:
        # list comprehension (генератор списков)
        values = [int(arg) for arg in args]

        n = len(values)
        i, p = 1, 1
        for i, value in enumerate(values, 0):
            p *= value
        g = pow(p, (1/(n)))
        g = '{:.4f}'.format(g)
        return g
    else:
        return None

if __name__ == "__main__":
    print('Введите через пробел аргументы для расчетов')
    ar = input().split()
    print(G(*ar))
```

```
C:\Users\Valentina\AppData\Local\Programs\Python\Python38-32\python.exe
Введите через пробел аргументы для расчетов
1 5 8 4 3 9
4.0357

Process finished with exit code 0
|
```

Рисунок 2 – Задание 1

Задание 2 (рис. 4-5):

Написать функцию, вычисляющую среднее гармоническое своих аргументов. Если функции передается пустой список аргументов, то она должна вернуть значение None.

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

Рисунок 4 – Среднее гармоническое

Код:

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

def G(*args):
    if args:
        # list comprehension (генератор списков)
        values = [int(arg) for arg in args]

        n = len(values)
        i, summ = 0, 0
        for i, value in enumerate(values, 0):
            summ += 1/value
        garm = n/summ
        garm = '{:.4f}'.format(garm)
        return garm
    else:
        return None

if __name__ == "__main__":
    print('Введите через пробел аргументы для расчетов')
    ar = input().split()
    print(G(*ar))
```

```
C:\Users\Valentina\AppData\Local\Programs\Python\
Введите через пробел аргументы для расчетов
20 50
28.5714

Process finished with exit code 0
```

Рисунок 5 – Задание 2

Задание 4. Вариант (рис. 7):

Самостоятельно подобрать или придумать задачу с переменным числом именованных аргументов. Привести решение этой задачи.

Задан список котов и их вес в килограммах. Необходимо определить самого крупного кота и средний вес каждого.

Код:

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

def cat(**kwargs):
    sum, sr = 0, 0
```

```

n = len(kwargs)
max = 0
for items, value in kwargs.items():
    sum += kwargs[items]
    if kwargs[items] > max:
        max = kwargs[items]
        name = items
sr = sum / n
sr = '{:.2f}'.format(sr)

print(f'Самый толстый кот - {name} - весит {max} кг')
print(f'В среднем каждый кот весит {sr} кг')

if __name__ == '__main__':
    cat(Афина=7.1, Мафин=12.4, Том=10.8,
        Финик=2.4, Семен=8.9, Персик=3.5, Марго=6.7)

```

```

C:\Users\Valentina\AppData\Local\Programs\Python\Python39\python.exe
Самый толстый кот - Мафин - весит 12.4 кг
В среднем каждый кот весит 7.40 кг

Process finished with exit code 0
|

```

Рисунок 6 – Задание 3

Задание 4. Вариант (рис. 7):

Написать функцию, принимающую произвольное количество аргументов, и возвращающую требуемое значение. Если функции передается пустой список аргументов, то она должна возвращать значение None.

Сумму аргументов, расположенных после первого положительного аргумента.

В процессе решения не использовать преобразования конструкции *args в список или иную структуру данных.

Код:

```

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

def G(*args):
    if args:
        summ = 0
        for i, value in enumerate(args):
            if int(value) > 0:
                for value2 in args[i + 1:]:
                    summ += int(value2)
                return summ
    else:
        return None

if __name__ == "__main__":
    print('Введите через прбел аргументы для расчетов')

```

```
ar = list(input().split())
print(G(*ar))
```

```
C:\Users\Valentina\AppData\Local\Programs\Python\Python38\python.exe
Введите через пробел аргументы для расчетов
-5 -6 7 4 -3 -3
-2
Process finished with exit code 0
```

Рисунок 7 – Задание 4

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: при выполнении заданий были приобретены навыки по работе с функциями с переменным числом аргументов при написании программ с помощью языка программирования Python.