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

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 12

Функции с переменным числом параметров в языке Python

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1
Злыгостев И.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 3.x.

Ход работы

1. После изучения теоретической части методических указаний, приступил к выполнению общих заданий.

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


def composition(*args):
    a = 1
    numbers = [float(arg) for arg in args]
    numbers.sort()
    for i in numbers:
        a *= i
    a = a ** (1/len(numbers))
    return a


def main(arg):
    if arg == '':
        print(None)
    else:
        arg = arg.split(',')
        args = {}
        for i, n in enumerate(arg):
            args[i] = int(n)
        print("Среднее геометрическое:", composition(*args.values()))


if __name__ == '__main__':
    arg = input('Введите список аргументов через запятую: ')
    main(arg)
```

Рисунок 12.1 – Код первого общего задания

```
Введите список аргументов через запятую: 4,2,1,5,6
Среднее геометрическое: 2.9925557394776896
```

Рисунок 12.2 – Первый результат выполнения кода

```
Введите список аргументов через запятую:
None
```

Рисунок 12.3 – Второй результат выполнения кода

```
Введите список аргументов через запятую: 5,1,4,9,0,23
Среднее геометрическое: 0.0
```

Рисунок 12.4 – Третий результат выполнения кода

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


def harmonic_mean_of_numbers(*args):
    summ = 0
    numbers = [float(arg) for arg in args]
    numbers.sort()
    for i in numbers:
        summ += 1/i
    harmonic = 1/(1/len(numbers)*summ)
    return harmonic


def main(arg):
    if arg == '':
        print(None)
    else:
        arg = arg.split(',')
        args = {}
        for i, n in enumerate(arg):
            args[i] = int(n)
        print("Среднее гармоническое элементов: ",
              harmonic_mean_of_numbers(*args.values()))


if __name__ == '__main__':
    arg = input('Введите список аргументов через запятую: ')
    main(arg)
```

Рисунок 12.5 – Код второго общего задания

```
Введите список аргументов через запятую: 1,4,5,7,4
Среднее гармоническое элементов:  2.7131782945736434
```

Рисунок 12.6 – Первый результат выполнения кода

```
Введите список аргументов через запятую:
None
```

Рисунок 12.7 – Второй результат выполнения кода

```
Введите список аргументов через запятую: 4,6,1,8,2
Среднее гармоническое элементов:  2.4489795918367347
```

Рисунок 12.8 – Третий результат выполнения кода

Индивидуальное задание

Вариант 5

2. После выполнения общих заданий приступил к работе с индивидуальным.

```python
def positive(*arguments):
    summ = 0
    last = 0
    numbers = [int(argument) for argument in arguments]
    for i, n in enumerate(numbers):
        if i >= last and n > 0:
            last = i
    arguments = numbers[:last]
    for i in arguments:
        summ += i
    return summ


def main(argument):
    if argument == '':
        print(None)
    else:
        argument = argument.split(',')
        arguments = {}
        for i, n in enumerate(argument):
            arguments[i] = int(n)
        print("Сумма аргументов, расположенных до последнего"
              " положительного числа:",
              positive(*arguments.values()))


if __name__ == '__main__':
    argument = input('Введите список аргументов через запятую: ')
    main(argument)
```

Рисунок 12.9 – Код индивидуального задания

```
Введите список аргументов через запятую: 4,1,3,0,5,1,-6,4
Сумма аргументов, расположенных до последнего положительного числа: 8
```

Рисунок 12.10 – Результат выполнения кода

3. Затем придумал собственное задание: “С клавиатуры вводится последовательность чисел. Найти среднее арифметическое этой последовательности”. И приложил одно из решений к этому заданию.

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


def the_amount(*arguments):
    summ = 0
    numbers = [float(arg) for arg in arguments]
    numbers.sort()
    for i in numbers:
        summ += i
    summ = summ/len(numbers)
    return summ


def main(argument):
    if argument == '':
        print(None)
    else:
        argument = argument.split(',')
        arguments = {}
        for i, n in enumerate(argument):
            arguments[i] = int(n)
        print("Среднее арифметическое:", the_amount(*arguments.values()))


if __name__ == '__main__':
    argument = input('Введите список аргументов через запятую: ')
    main(argument)
```

Рисунок 12.11 – Код придуманного задания

```
Введите список аргументов через запятую: 4,1,7,1,2,3
Среднее арифметическое: 3.0
```

Рисунок 12.12 – Результат выполнения кода

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