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 алгоритмизация»

Выполнил студент группы ИВТ-б-о-20-1

Галяс Д. И. « »_____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:

Ссылка на репозиторий: https://github.com/DIMITRY-GALYAS1/Laba-2.10.git

1. Создал новый репозиторий на github, после клонировал его и создал в папке репозитория новый проект PyCharm.

2. Выполнил пример.

```
C:\Users\lizeq\anaconda3\envs\pythonProject5\python.exe C:/Users/lizeq/PycharmProjects/pythonProject5/primer_1.py
None
6.0
4.5
```

Рисунок 1. Пример

3. Выполнил первое задание.

```
"""
Решить поставленную задачу:
написать функцию, вычисляющую среднее геометрическое
своих аргументов a1, a2, ... an
Если функции передается пустой список аргументов,
то она должна возвращать значение  None
"""


# Вычисляет среднее геометрическое
def average(*arg):

    if arg:
        g = 1.0

        for i in arg:
            g *= i
        g = g ** (1/len(arg))

        return g
    else:
        return None


if __name__ == '__main__':

    print("Введите числа в массив через пробел: ")
    mas = list(map(float, input().split()))
    print(average(*mas))
```

Рисунок 2. Код первого задание

```
C:\Users\lizeq\anaconda3\envs\pythonProject5\python.exe C:/Users/lizeq/PycharmProjects/pythonProject5/zadanie_1.py
Введите числа в массив через пробел:
2 4 5 5
3.7606030930863934
```

Рисунок 3. Выполнение первого задания

4. Выполнил второе задание.

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


"""
Решить поставленную задачу:
написать функцию, вычисляющую среднее гармоническое
своих аргументов a1, a2, ... an
Если функции передается пустой список аргументов,
то она должна возвращать значение  None
"""


def average(*x):
    summa = 0
    for i in x:
        if i == 0:
            return 'В введенном списке есть 0'
        else:
            summa += 1 / float(i)
    z = 1 / (1 / len(x) * summa)
    return z


if __name__ == '__main__':

    print("Введите числа в массив через пробел (нули запрещены): ")
    mas = list(map(float, input().split()))
    print(average(*mas))
```

Рисунок 4. Код второго задания

```
C:\Users\lizeq\anaconda3\envs\pythonProject5\python.exe C:/Users/lizeq/PycharmProjects/pythonProject5/zadanie_2.py
Введите числа в массив через пробел:
4 5 5 5 2
3.7037037037037033
```

Рисунок 5. Выполнение второго задания

5. Выполнил третье задание.

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


"""
Дан список учащихся 5 "а" класса, в котором указаны фамилии и рост.
Найти рост самого высокого ученика.
"""


def school(**most):
    maximum = 0
    z = ""
    for name, height in most.items():
        if height > maximum:
            maximum = height
            z = name
    print(f"Самый высокий ученик {z} имеет рост {maximum} см")


if __name__ == '__main__':
    school(Иван=150, Коля=167, Дима=186, Игорь=177)
```

```
school()
zadanie_3
C:\Users\lizeq\anaconda3\envs\pythonProject5\python.exe C:/Users/lizeq/PycharmProjects/pythonProject5/zadanie_3.py
Самый высокий ученик Дима имеет рост 186 см

Process finished with exit code 0
```

Рисунок 6. Код третьего задания

6. Выполнил индивидуальное задание.

```python
"""
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
"""


def between(*args):

    for i, x in enumerate(args):
        if x == 0:
            z = 0
            for meaning in args[i + 1:]:
                if meaning == 0:
                    break
                else:
                    z += meaning
            return z


if __name__ == '__main__':
    print("Введите числа в массив через пробел: ")
    q = list(map(float, input().split()))
    print(between(*q))
```

Рисунок 7. Код индивидуального задания

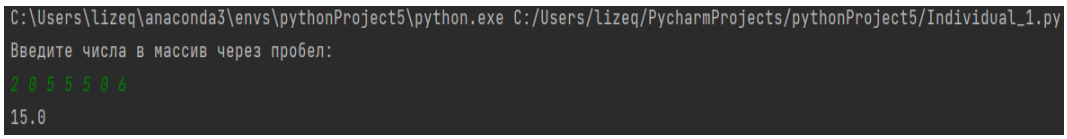

Рисунок 8. Выполнение индивидуального задания

Контрольные вопросы:

1. Какие аргументы называются позиционными в Python?

При вызове функции аргументы можно передавать как позиционные – передаются в том же порядке, в котором они определены при создании функции. То есть, порядок передачи аргументов определяет, какое значение получит каждый аргумент.

2. Какие аргументы называются именованными в Python?

Аргументы, передаваемые с именами, называются именованными. При вызове функции можно использовать имена параметров из ее определения. Благодаря **kwargs создаётся словарь, в котором содержатся именованные аргументы, переданные функции при её вызове.

3. Для чего используется оператор *?

Этот оператор позволяет «распаковывать» объекты, внутри которых хранятся некие элементы.

```
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?

*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.

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