МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

«Функции с переменным числом параметров в Python»

Отчет по лабораторной работе № 2.10

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-21-1

Коновалова В.Н. « » 2022г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

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. Изучить теоретический материал работы.

2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия IT и язык программирования Python.

3. Выполните клонирование созданного репозитория.

4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.

5. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

6. Создайте проект PyCharm в папке репозитория.

7. Проработайте пример лабораторной работы.

Пример 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))
```

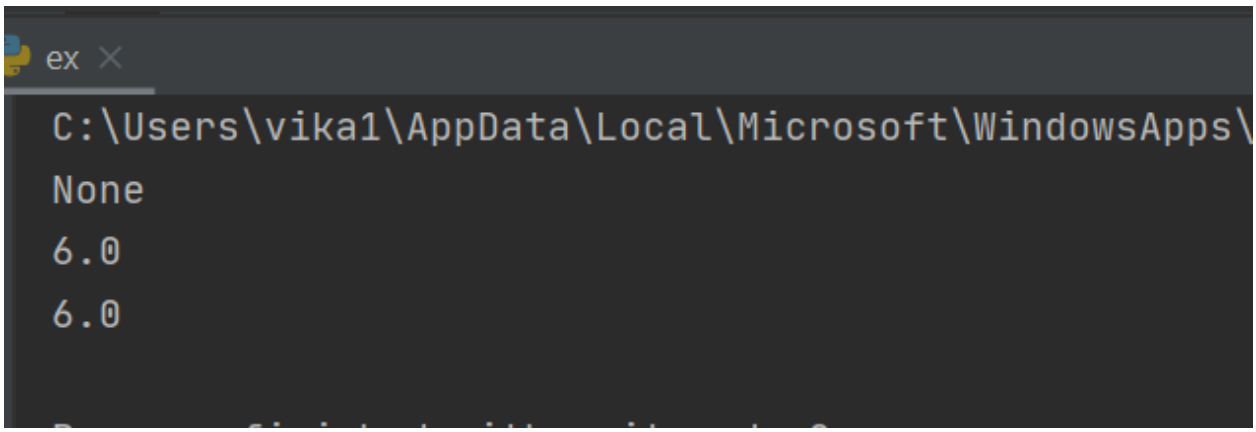

Рисунок 6 – Результат работы программы

8. Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов $a_1, a_2, \ldots, a_n$

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

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

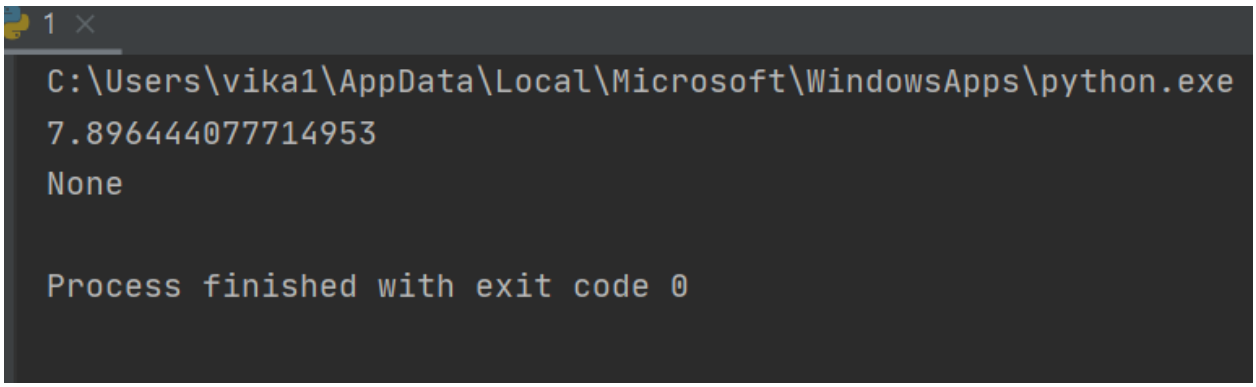

Рисунок 6 – Результат работы программы

9. Решить поставленную задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов $a_1, a_2, \ldots, a_n$

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

Код:

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

def sred_garmoni(*args):
    if args:
        values = [float(arg) for arg in args]
        n = len(values)
        a = 0
        for i in values:
            a = a + (1 / i)
        return n / a
    else:
        return None


if __name__ == "__main__":
    print(sred_garmoni())
    print(sred_garmoni(4, 7, 13, 21))
```

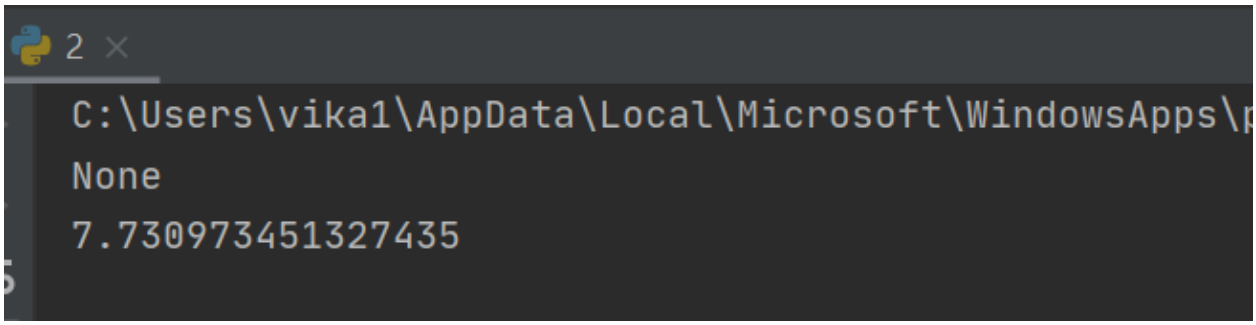

Рисунок 7 – Результат работы программы

10. Зафиксируйте сделанные изменения в репозитории.

11. Решите индивидуальное задание согласно своему варианту.

Код:

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

import math


def mult_after_max(*args):
    """
    Произведение аргументов, расположенных после
    максимальногопо модулю аргумента.
    """
    if args:
        mult = 1
        values = [float(arg) for arg in args]
        max_item = 0
        max_ind = 0
        for i, item in enumerate(values):
            if math.fabs(item) > max_item:
                max_ind = i
                max_item = math.fabs(item)
        values = values[max_ind:]
        for arg in values:
            mult = mult * arg
        return mult
    else:
        return None


if __name__ == "__main__":
    print(f'Произведение: {mult_after_max()}')
    print(f'Произведение: {mult_after_max(1, 3, 6, 4, 5)}')
    print(f'Произведение: {mult_after_max(7, 8, 12, 76, 34, 7, 34)}')
```

```
Произведение: None
Произведение: 120.0
Произведение: 614992.0
```

Рисунок 9 – Результат работы программы

12. Зафиксируйте сделанные изменения в репозитории.

13. Самостоятельно подберите или придумайте задачу с переменным числом именованных аргументов. Приведите решение этой задачи.

Код:

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

```python
"""
Напишите функцию, принимающую произвольное количество аргументов, и
возвращающую
требуемое значение. Если функции передается пустой список аргументов, то она
должна
возвращать значение None . Номер варианта определяется по согласованию с
преподавателем.
В процессе решения не использовать преобразования конструкции *args в список
или иную
структуру данных.
8. Сумму аргументов, расположенных между первым и последним положительными
аргументами.
"""


def sump(*args):
    """
    Находит сумму аргументов расположенных между первым и последним
положительным аргументами
    """
    s1 = s2 = 0
    if args:
        for s1, a in enumerate(args):
            if a > 0:
                break  # определить индекс первого положительного элемента
        for s2, a in enumerate(reversed(args)):
            if a > 0:
                break  # определить индекс последнего положительного элемента
        return sum(args[s1 + 1: -s2 - 1])  # взять сумму между индексами
    else:
        return None


if __name__ == "__main__":
    print(sump())
    print(sump(-3, 5, -7, -12, 13, 18))
```

```
id ×
C:\Users\vika1\AppData\Local\Microsoft\WindowsApps\python
None
-6

Process finished with exit code 0
```

Рисунок 11 – Результат работы программы

12. Зафиксируйте сделанные изменения в репозитории.

Вопросы для защиты работы

1. Какие аргументы называются позиционными в Python?

Это аргументы, передаваемые в вызов в определённой последовательности (на определённых позициях), без указания их имён. Элементы объектов, поддерживающих итерирование, могут использоваться в качестве позиционных аргументов, если их распаковать при помощи *.

2. Какие аргументы называются именованными в Python?

Это аргументы, передаваемые в вызов при помощи имени (идентификатора), либо словаря с его распаковкой при помощи **.

3. Для чего используется оператор *?

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

4. Каково назначение конструкций *args и **kwargs?

*args используется для передачи произвольного числа неименованных аргументов функции.

**kwargs позволяет передавать произвольное число именованных аргументов в функцию.