

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

**«Лабораторная работа 2.10 Функции с
переменным числом параметров в
Python»**

**ОТЧЕТ
по лабораторной работе №13
дисциплины
«Основы программной инженерии»**

Выполнил:

Луценко Дмитрий Андреевич
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Лабораторная работа 2.10 Функции с переменным числом параметров в Python

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:

Задание 1

Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов a_1, a_2, \dots, a_n

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

```
1  ▶  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import math
5
6
7  DFooRS
8  def aver_geom(*args):
9      if args:
10         mult = 1
11         n = 0
12         for item in args:
13             mult *= item
14             n += 1
15         n = 1 / n
16     return math.pow(mult, n)
17
18     else:
19         return None
20
21
22  ▶  if __name__ == "__main__":
23         print(aver_geom(5, 4, 2))
```

Рисунок 1 – Код программы для задания 1

```
22 ▶ if __name__ == "__main__":
23     print(aver_geom(5, 4, 2))
24
aver_geom() > if args
1task ×
E:\GitHub\13lab\user\Scripts\python.exe
3.4199518933533937
```

Рисунок 2 – Результат работы программы

Задание 2

Решить поставленную задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов a_1, a_2, \dots, a_n

$$\frac{n}{H} = \sum_{k=1}^n \frac{1}{a_k}. \quad (2)$$

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def aver_garm(*args):
6     if args:
7         n = 0
8         sum = 0
9         for item in args:
10             sum += 1 / item
11             n += 1
12
13         return n / sum
14
15     else:
16         return None
17
18
19 ▶ if __name__ == "__main__":
20     print(aver_garm(5, 4, 2))
```

Рисунок 3 – Код программы для задания 2

```
19 ▶ if __name__ == "__main__":
20     print(aver_garm(5, 4, 2, 7, 3))

aver_garm() > else

2task x

E:\GitHub\13laba\user\Scripts\python.exe E:\Gi
3.505843071786311
```

Рисунок 4 – Результат работы программы

Индивидуальное задание

18. Сумму положительных аргументов, расположенных до максимального аргумента.

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def sum_elem(*args):
6     if args:
7         values = tuple(float(arg) for arg in args)
8
9         sum_num = 0
10        max_num = 0
11        for i, item in enumerate(values):
12            if item > max_num:
13                n = i
14                max_num = item
15        for item in values[:n]:
16            if item > 0:
17                sum_num += item
18
19        return sum_num
20
21    else:
22        return None
```

Рисунок 5 – Код программы для индивидуального задания

```
24
25 ► if __name__ == "__main__":
26     print(sum_elem(3, -2, 1, 5, 4, 43, 9, -4, 7))

sum_elem()
individual x
↑ E:\GitHub\13laba\user\Scripts\python.exe E:\GitHub\13laba\in
↓ 13.0
```

Рисунок 6 – Результат работы программы

Вывод: были приобретены навыки по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.

Ответы на контрольные вопросы

1. Какие аргументы называются позиционными в Python?

Позиционные аргументы – это аргументы вызванные при помощи оператора*

2. Какие аргументы называются именованными в Python?

Именованные аргументы – это аргументы, вызванные при помощи оператора**.

3. Для чего используется оператор * ? Оператор «звёздочка» в Python способен «вытаскивать» из объектов составляющие их элементы.

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.