

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №13
дисциплины «Программирование на Python»
Вариант 5.

Выполнила:
Михеева Елена Александровна
2 курс, группа ИВТ-б-з-20-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: 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.

Порядок выполнения работы.

1. Был проработан пример лабораторной работы. В отчете представлены скриншот работы программы примера.

```
None
6.0
4.5
(base) mikheeva@MacBook-Pro-Elena Python_2.10 %
```

Рисунок 1. Выполнение программы примера 1

2. Было выполнено 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
5  def calculating_multiply(*args):
6      """
7      Вычисление среднего геометрического полученных аргументов.
8      """
9
10     if args:
11         multiply = 1.0
12         for value in args:
13             multiply *= value
14         return round(pow(multiply, 1/len(args)), 2)
15     else:
16         return "None"
17
18
19  if __name__ == "__main__":
20     elements = [1, 2, 8, 9, 2]
21     print("Elements:", *elements)
22     print("Geometric mean of elements: ", calculating_multiply(*elements))
23     print(calculating_multiply())
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Elements are: 1 2 8 9 2
Answer is - 3.1
None
(base) mikheeva@MacBook-Pro-Elena Python_2.10 %
```

Рисунок 3. Программа для задания №1

3. Было выполнено 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 calculating_garmonical(*args):
6      """
7      Вычисление среднего гармонического полученных аргументов.
8      """
9
10     if args:
11         garmony = 0
12         for value in args:
13             garmony += 1 / value
14         return round(len(args)/garmony, 2)
15     else:
16         return "None"
17
18
19 if __name__ == "__main__":
20     elements = [1, 2, 8, 9, 2]
21     print("Elements: ", *elements)
22     print("Harmonic mean of elements: ", calculating_garmonical(*elements))
23     print(calculating_garmonical())
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Elements: 1 2 8 9 2
Harmonic mean of elements: 2.24
None
(base) mikheeva@MacBook-Pro-Elena Python_2.10 %
```

Рисунок 4. Программа для задания №2

4. Приступили к выполнению индивидуального задания согласно варианту №5.

Напишите функцию, принимающую произвольное количество аргументов, и возвращающую требуемое значение. 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.

Найти сумму аргументов, расположенных до последнего положительного аргумента.

```
5 def sum_before_last_positive(*args):
6     """
7     Функция находит сумму аргументов,
8     расположенных до последнего положительного аргумента.
9     """
10    if not args:
11        return None
12
13    last_positive_index = -1
14
15    for i, arg in enumerate(args):
16        if arg > 0:
17            last_positive_index = i
18
19    if last_positive_index == -1:
20        return None
21
22    return sum(args[:last_positive_index])
23
24
25 if __name__ == "__main__":
26     result = sum_before_last_positive(1, 3, -2, 3, 4, -5, 6)
27     print("Sum before last positive elements ", result)
28
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Sum before last positive elements 4
(base) mikheeva@MacBook-Pro-Elena Python_2.10 %

Рисунок 5. Программа индивидуального задания

Ответы на контрольные вопросы.

1. Какие аргументы называются позиционными в Python?

Позиционные аргументы в Python – это аргументы, которые передаются в функцию в определенном порядке. Они соответствуют порядку параметров в определении функции.

2. Какие аргументы называются именованными в Python?

Именованные аргументы в Python - это аргументы, которые передаются в функцию с указанием их имени. Они позволяют передавать аргументы в любом порядке.

3. Для чего используется оператор * ?

Оператор * в Python используется для распаковки итерируемых объектов, таких как списки или кортежи, в аргументы функции. Он также может быть использован для распаковки словарей в именованные аргументы функции.

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

`*args` и `**kwargs` – это общепринятые названия для аргументов функций, которые позволяют передавать переменное количество позиционных и именованных аргументов соответственно. `*args` используется для передачи переменного количества позиционных аргументов, а `**kwargs` - для передачи переменного количества именованных аргументов.