

12Министерство 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»

Выполнил:
Кожуховский Виктор Андреевич
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных систем
», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

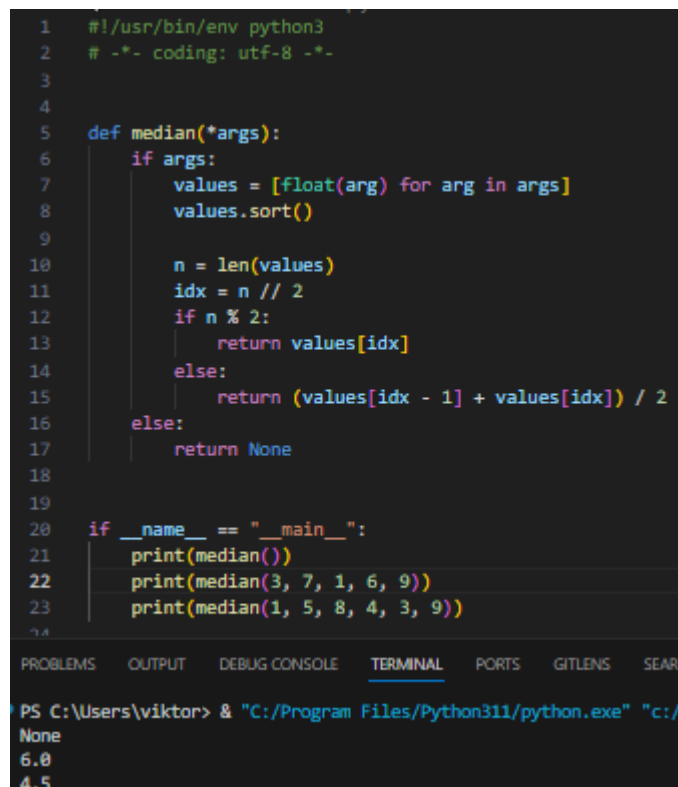
Ставрополь, 2023 г.

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.

Методика и порядок выполнения работы

1. Изучил теоретический материал работы.
2. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.
3. Выполнил клонирование созданного репозитория.
4. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организовал свой репозиторий в соответствии с моделью ветвления git-flow.
6. Создал проект в папке репозитория.
7. Проработал пример лабораторной работы.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def median(*args):
6      if args:
7          values = [float(arg) for arg in args]
8          values.sort()
9
10         n = len(values)
11         idx = n // 2
12         if n % 2:
13             return values[idx]
14         else:
15             return (values[idx - 1] + values[idx]) / 2
16     else:
17         return None
18
19
20 if __name__ == "__main__":
21     print(median())
22     print(median(3, 7, 1, 6, 9))
23     print(median(1, 5, 8, 4, 3, 9))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SEARCH

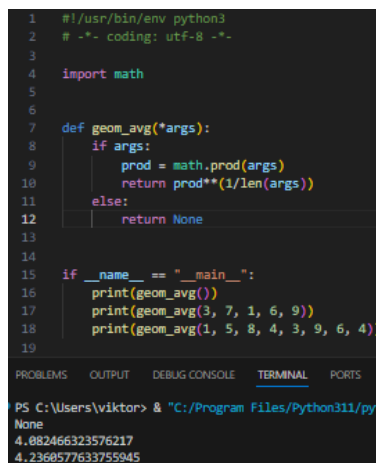
```
PS C:\Users\viktor> & "C:/Program Files/Python311/python.exe" "c:/A
None
6.0
4.5
```

Рисунок 1. Код примера и его выполнение

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

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

Если функции передается пустой список аргументов, то она должна возвращать значение None.



```
1 #!/usr/bin/env python3
2 #- coding: utf-8 -*-
3
4 import math
5
6
7 def geom_avg(*args):
8     if args:
9         prod = math.prod(args)
10        return prod**(1/len(args))
11    else:
12        return None
13
14
15 if __name__ == "__main__":
16     print(geom_avg())
17     print(geom_avg(3, 7, 1, 6, 9))
18     print(geom_avg(1, 5, 8, 4, 3, 9, 6, 4))
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

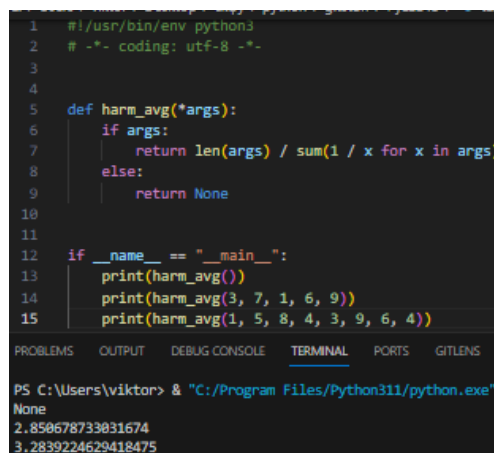
PS C:\Users\viktor> & "C:/Program Files/Python311/py
None
4.082466323576217
4.2360577633755945

Рисунок 2. Код решения задачи 1 и его выполнение

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

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

Если функции передается пустой список аргументов, то она должна возвращать значение None.



```
1 #!/usr/bin/env python3
2 #- coding: utf-8 -*-
3
4
5 def harm_avg(*args):
6     if args:
7         return len(args) / sum(1 / x for x in args)
8     else:
9         return None
10
11
12 if __name__ == "__main__":
13     print(harm_avg())
14     print(harm_avg(3, 7, 1, 6, 9))
15     print(harm_avg(1, 5, 8, 4, 3, 9, 6, 4))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

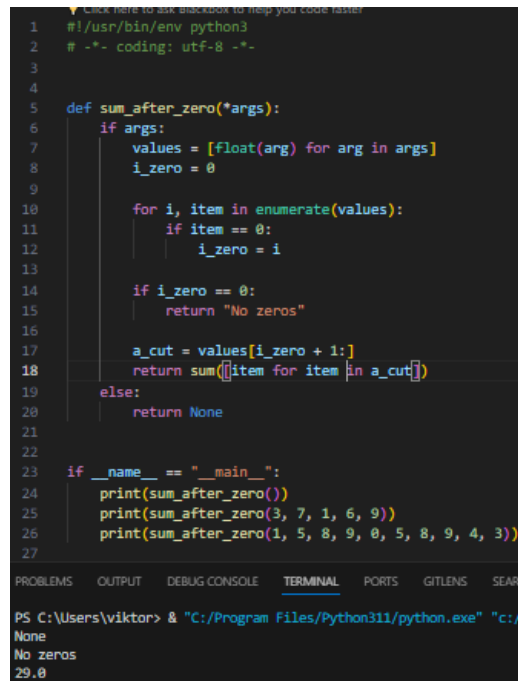
PS C:\Users\viktor> & "C:/Program Files/Python311/python.exe"
None
2.850678733031674
3.2839224629418475

Рисунок 3. Код решения задачи 2 и его выполнение

11. Зафиксировал сделанные изменения в репозитории.

12. Решил индивидуальное задание согласно своему варианту.

Сумму аргументов, расположенных после последнего аргумента, равного нулю.



```
1  Click here to ask blackbox to help you code faster
2  #!/usr/bin/env python3
3  # -*- coding: utf-8 -*-
4
5  def sum_after_zero(*args):
6      if args:
7          values = [float(arg) for arg in args]
8          i_zero = 0
9
10         for i, item in enumerate(values):
11             if item == 0:
12                 i_zero = i
13
14         if i_zero == 0:
15             return "No zeros"
16
17         a_cut = values[i_zero + 1:]
18         return sum([item for item in a_cut])
19     else:
20         return None
21
22
23 if __name__ == "__main__":
24     print(sum_after_zero())
25     print(sum_after_zero(3, 7, 1, 6, 9))
26     print(sum_after_zero(1, 5, 8, 9, 0, 5, 8, 9, 4, 3))
27
```

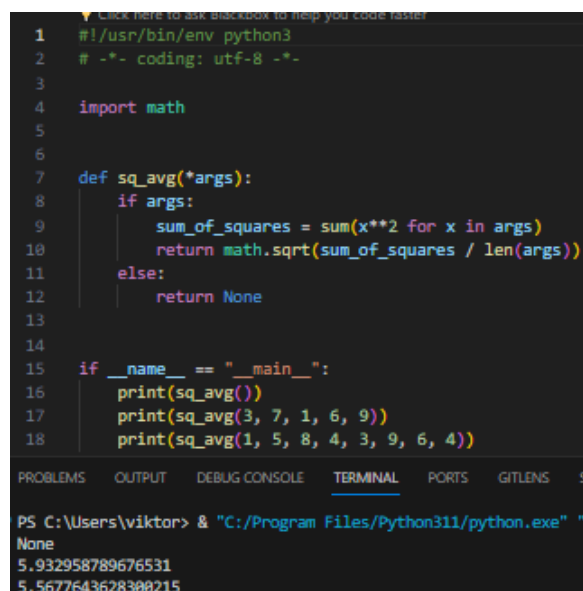
PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS GITLENS SEARCH

```
PS C:\Users\viktor> & "C:/Program Files/Python311/python.exe" "c:/U
None
No zeros
29.0
```

Рисунок 4. Код решения индивидуального задания и его выполнение

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

Вычислить среднее квадратическое.



```
1  Click here to ask blackbox to help you code faster
2  #!/usr/bin/env python3
3  # -*- coding: utf-8 -*-
4
5  import math
6
7  def sq_avg(*args):
8      if args:
9          sum_of_squares = sum(x**2 for x in args)
10         return math.sqrt(sum_of_squares / len(args))
11     else:
12         return None
13
14
15 if __name__ == "__main__":
16     print(sq_avg())
17     print(sq_avg(3, 7, 1, 6, 9))
18     print(sq_avg(1, 5, 8, 4, 3, 9, 6, 4))
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS GITLENS SE

```
PS C:\Users\viktor> & "C:/Program Files/Python311/python.exe" "c
None
5.932958789676531
5.5677643628308215
```

Рисунок 5. Код решения придуманной задачи и его выполнение

14. Зафиксировал сделанные изменения в репозитории.
15. Добавил отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксировал изменения.
16. Выполнил слияние ветки для разработки с веткой master / main.
17. Отправил сделанные изменения на сервер GitHub.

Вопросы для защиты работы

1. Какие аргументы называются позиционными в Python?

Позиционные аргументы - это аргументы, которые передаются в вызове функции в порядке, который был определен при объявлении функции. Их значения связываются с параметрами функции исходя из их позиции.

2. Какие аргументы называются именованными в Python?

Именованные аргументы - это аргументы, передающиеся в функцию с использованием имени параметра, к которому они должны быть присвоены. Это позволяет задавать их в любом порядке, не следуя порядку параметров в определении функции.

3. Для чего используется оператор *?

Оператор * используется для распаковки итерируемых объектов и передачи их элементов как отдельных позиционных аргументов в функцию. Также используется в определении функций для сбора произвольного количества позиционных аргументов в один параметр в виде кортежа.

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

*args — это сокращение от «arguments» (аргументы), а **kwargs — сокращение от «keyword arguments» (именованные аргументы).