

МИНЕСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ
по лабораторной работе №5
Дисциплина: «Языки программирования»

Выполнил: студент 2 курса
группы ИТС-б-о-20-1

Попов Данила Владимирович

Проверил:

к.ф.-м.н., доцент

кафедры инфокоммуникаций

Воронкин Роман Александрович

Работа защищена с оценкой: _____

Ставрополь, 2022

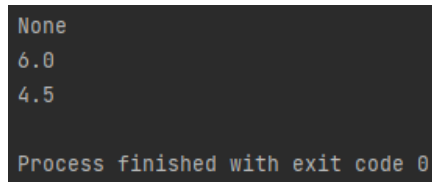
Функции с переменным числом параметров в Python

Цель работы: приобретение навыков по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.

<https://github.com/12W300/Five.git>

Ход работы:

Пример. Разработать функцию для определения медианы значений аргументов функции. Если функции передается пустой список аргументов, то она должна возвращать значение None.



```
None
6.0
4.5

Process finished with exit code 0
```

Рисунок 1 – результат работы примера

Задание 1. Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов. Если функции передается пустой список аргументов, то она должна возвращать значение None.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5
6      def midlgeom(a):
7          if len(a) != 0:
8              res = 1
9              for i in range(len(a)):
10                 res *= a[i]
11                 return pow(res, 1 / (len(a)))
12             else:
13                 return None
14
15     raw = input('Введите последовательность чисел через пробел: ')
16     mas = [int(i) for i in raw.split(' ') if i.isdigit()]
17
18     print(midlgeom(mas))
19
```

Рисунок 2 – код задания 1

```
Введите последовательность чисел через пробел: 1 3 5 7 9
3.936283427035352

Process finished with exit code 0
```

Рисунок 3 – результат работы кода

Задание 2. Решить поставленную задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов. Если функции передается пустой список аргументов, то она должна возвращать значение None.

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5
6     def midlgeom(a):
7         if len(a) != 0:
8             res = 0
9             for i in range(len(a)):
10                 res += 1/a[i]
11             return len(a) / res
12         else:
13             return None
14
15     raw = input('Введите последовательность чисел через пробел: ')
16     mas = [int(i) for i in raw.split(' ') if i.isdigit()]
17
18     print(midlgeom(mas))
19
```

Рисунок 4 – код задания 2

```
Введите последовательность чисел через пробел: 1 3 5 7 9
2.797513321492007

Process finished with exit code 0
```

Рисунок 5 – результат работы кода

Индивидуальное задание варианта 13. Напишите функцию, принимающую произвольное количество аргументов, и возвращающую сумму аргументов, расположенных после минимального аргумента. 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.

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5
6     def summ(*args):
7         if args:
8             values = [int(arg) for arg in args]
9             print(values)
10            imin = 0
11            minval = int(values[0])
12            for i in range(len(values)):
13                if values[i] < minval:
14                    minval = values[i]
15                    imin = i
16            imin += 1
17            sum = 0
18            for i in range(imin, len(values)):
19                sum += abs(values[i])
20            return sum
21        else:
22            return None
23
24    print("сумма ", summ(3, 7, 1, -3, 2, 6, 0, 5))
25
```

Рисунок 6 – код индивидуального задания

```
[3, 7, 1, -3, 2, 6, 0, 5]
сумма 13

Process finished with exit code 0
```

Рисунок 7 – результат работы кода

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

```

1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5
6     def print_values(
7         cpu="Intel",
8         ram=16,
9         disk="SSD",
10        video="NVIDIA",
11    ):
12        print(f"Процессор {cpu}")
13        print(f"Память {ram}Gb")
14        print(f"Накопитель {disk}")
15        print(f"Видео {video}")
16
17    print('все значения по умолчанию')
18    print_values()
19
20    newPC1 = {"cpu": "AMD"}
21    print('изменяем процессор для PC1')
22    print_values(**newPC1)
23
24    newPC2 = {"video": "Radeon", "ram": 64}
25    print('изменяем видеокарту и память для PC2')
26    print_values(**newPC2)
27
28    newPC3 = {"cpu": "Ryzen", "ram": 128}
29    print('изменяем процессор и память для PC2')
30    print_values(**newPC3)
31

```

Рисунок 8 – код 2 индивидуального задания

```

все значения по умолчанию
Процессор Intel
Память 16Gb
Накопитель SSD
Видео NVIDIA
изменяем процессор для PC1
Процессор AMD
Память 16Gb
Накопитель SSD
Видео NVIDIA
изменяем процессор и память для PC2
Процессор Ryzen
Память 128Gb
Накопитель SSD
Видео NVIDIA

```

Рисунок 9 – результат работы кода

Контрольные вопросы:

1. Какие аргументы называются позиционными в Python?

Это аргументы, передаваемые в вызов в определённой последовательности (на определённых позициях), без указания их имён с распаковкой 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`?

Каждая из этих конструкций используется для распаковки аргументов соответствующего типа, позволяя вызывать функции со списком аргументов переменной длины.

При использовании `**kwargs` создаётся словарь, в котором содержатся именованные аргументы, переданные функции при её вызове.

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