

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 ПРАКТИЧЕСКОЙ РАБОТЕ №8**  
**дисциплины «Программирование на Python»**

Выполнил:

Лейс Алексей Вячеславович  
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»

---

(подпись)

Руководитель практики: кандидат тех.  
наук доцент кафедры  
инфокоммуникаций: Воронкин Р.А

---

(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:

Пример из лабораторной работы функции с переменной

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

### Индивидуальное задание:

Произведение аргументов, расположенных между максимальным и минимальным аргументами.

```
15 # Определяем начальный и конечный индексы для расчета произведения
16 start_index = min(max_index, min_index) + 1
17 end_index = max(max_index, min_index)
18
19 # Исключаем случай, когда начальный и конечный индексы совпадают
20 if start_index == end_index:
21     return None
22
23 # Вычисляем произведение аргументов
24 product = 1
25 for i in range(start_index, end_index):
26     product *= args[i]
27
28 return product
29
30 if __name__ == "__main__":
31     result = find_product_between_max_min(3, 5, 2, 8, 4, 10)
32     print("Произведение аргументов между максимальным и минимальным:", result)
33
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell  
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (<https://aka.ms/pscore6>)

PS I:\ИВТ-6-о-22-1\Программирование на Python\13\lab\_2.10> & 'D:\LAV\Files\Programs\VisualStudio\Share\AV\vscode\extensions\ms-python.python-2023.20.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy-2-1\Программирование на Python\13\lab\_2.10\2.py'

Произведение аргументов между максимальным и минимальным: 32

Ответы на вопросы:

1. **Какие аргументы называются позиционными в Python?**

- Позиционные аргументы - это аргументы, передаваемые в функцию в том порядке, в котором они объявлены. Значения этих аргументов присваиваются параметрам функции в соответствии с их порядковым номером.

2. **Какие аргументы называются именованными в Python?**

- Именованные аргументы - это аргументы, передаваемые в функцию с явным указанием имени параметра. Это позволяет передавать аргументы в произвольном порядке и указывать только те аргументы, которые нужны.

3. **Для чего используется оператор \* ?**

- Оператор \* в Python используется для распаковки и упаковки последовательностей. В контексте аргументов функции \*args используется для передачи переменного числа позиционных аргументов, а \*sequence - для распаковки последовательности при передаче аргументов.

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

- **\*args** и **\*\*kwargs** используются для обработки переменного числа аргументов в функции. **\*args** позволяет передавать произвольное количество позиционных аргументов, а **\*\*kwargs** - произвольное количество именованных аргументов. Вместо **args** и **kwargs** можно использовать любые идентификаторы, но соглашение об использовании данных названий является общепринятым.

Вывод: В ходе работы приобрел навыки по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.