

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 «Основы программной инженерии»

Выполнил:
Плугатырев Владислав Алексеевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: функции с переменным числом параметров 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 выполнения работы

1. Создал репозиторий.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner * BigLofanbl4 / **Repository name *** lab13

✔ lab13 is available.

Great repository names are short and memorable. Need inspiration? How about [literate-octo-winner](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: Python

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: MIT License

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set main as the default branch. Change the default name in your [settings](#).

(i) You are creating a public repository in your personal account.

[Create repository](#)

Рисунок 1.1 – Создание репозитория

2. Проработал пример из лабораторной работы.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def median(*args):
    if args:
        values = [float(arg) for arg in args]
        values.sort()
        n = len(values)
        idx = n // 2
        if n % 2:
            return values[idx]
        else:
            return (values[idx - 1] + values[idx]) / 2
    else:
        return None

if __name__ == "__main__":
    print(median())
    print(median(3, 7, 1, 6, 9, 10, 18, 100))
    print(median(1, 5, 8, 4, 3, 9, 5))
```

Рисунок 2.1 – Код примера

```
None
8.0
5.0
```

Рисунок 2.2 – Вывод программы из примера

3. Написать функцию, вычисляющую среднее геометрическое своих аргументов.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def geom_mean_val(*nums):
    if nums:
        prod = 1
        for i in nums:
            prod *= i
        return prod ** (1 / len(nums))
    else:
        return None

if __name__ == "__main__":
    print(geom_mean_val(2, 2, 4))
    print(geom_mean_val())
```

Рисунок 3.1 – Код программы

```
2.5198420997897464
None
```

Рисунок 3.2 – Вывод программы

4. Написать функцию, вычисляющую среднее гармоническое своих аргументов.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def avg_harmonic(*nums):
    if nums:
        res = 0
        for i in nums:
            res += 1 / i
        return len(nums) / res
    else:
        return None

if __name__ == "__main__":
    print(avg_harmonic(3, 2, 1))
    print(avg_harmonic())
```

Рисунок 4.1 – Код программы

```
1.6363636363636365
None
```

Рисунок 4.2 – Вывод программы

5. Индивидуальное задание: написать функцию, которая находит произведение аргументов, расположенных после максимального по модулю аргумента.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def prod(*nums):
    if nums:
        max_elem = max([abs(i) for i in nums])
        max_elem_index = [abs(i) for i in nums].index(max_elem)
        res = 1

        for i in nums[max_elem_index + 1 :]:
            res *= i
        return res
    else:
        return None

if __name__ == "__main__":
    print(prod(1, 2, 3, 100, 2, -105, 7, 4))
    print(prod())
```

Рисунок 5.1 – Код программы

```
28
None
```

Рисунок 5.2 – Вывод программы

Ответы на контрольные вопросы

1. Позиционные аргументы в Python - это аргументы, которые передаются в функцию в том порядке, в котором они указаны в вызове функции.
2. Именованные аргументы в Python - это аргументы, которые передаются в функцию с указанием их имени и значения в виде пары “имя=значение”.
3. Оператор `*` используется в Python для распаковки последовательности аргументов.
4. Конструкция `*args` в Python используется для передачи произвольного количества позиционных аргументов в функцию. Аналогично, `**kwargs` используется для передачи произвольного количества именованных аргументов в функцию.