

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.10

Функции с переменным числом параметров в Python
по дисциплине «Технологии программирования и алгоритмизация»

Выполнил

студент группы ИВТ-б-о-20-1

Дыбов Д.В. « » _____ 20__ г.

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2021

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

Ход работы

1. Создал новый репозиторий для лабораторной работы №2.10;
2. Клонировал созданный репозиторий на компьютер;
3. Создал новый PyCharm проект в папке репозитория;
4. Проработал пример:
5. Проверил пример на работоспособность;

```
None
6.0
4.5
Process finished with exit code 0
```

Рисунок 1 – Результат выполнения примера

6. Выполнил первую задачу;
7. Проверил первую задачу на работоспособность:

```
Введите несколько чисел: 8 2 6 17
6.355943655622531
Process finished with exit code 0
```

Рисунок 2 – Результат выполнения первой задачи

8. Выполнил вторую задачу;
9. Проверил вторую задачу на работоспособность:

```
Введите несколько чисел: 3 2 6 9 5
3.813559322033899
Process finished with exit code 0
```

Рисунок 3 – Результат выполнения второй задачи

10. Выполнил третью задачу;

11. Проверил третью задачу на работоспособность:

```
('f', 1)
('s', 2)
('t', 4)
('fr', 5)
('fv', 7)
('sx', 8)
('sv', 9)
('e', 12)
('n', 16)
('tn', 21)

Process finished with exit code 0
```

Рисунок 4 – Результат выполнения третьей задачи

12. Выполнил третью задачу;

13. Проверил третью задачу на работоспособность:

```
Введите несколько чисел: 2 4 2 1 -8 3 6 8 1 9 -1 3 4 8
1296.0

Process finished with exit code 0
```

Рисунок 5 – Результат выполнения индивидуального задания

14. С помощью сайта проверил пример на наличие ошибок;

15. Результат не выдал ошибок;

Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def median(*args):
5     if args:
6         values = [float(arg) for arg in args]
7         values.sort()
8
9         n = len(values)
10        idx = n // 2
11        if n % 2:
12            return values[idx]
13        else:
```

No syntax errors detected :)

Рисунок 6 – Проверка кода на наличие ошибок

16. С помощью сайта проверил первую задачу на наличие ошибок;

17. Результат не выдал ошибок;

Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def geometric_mean(*args):
6     if args:
7         multi = 1
8         for i in args:
9             multi *= i
10        sr_geom = pow(multi, 1/len(args))
11        return float(sr_geom)
12
13     else:
```

No syntax errors detected :)

Рисунок 7 – Проверка первой задачи на наличие ошибок

18. С помощью сайта проверил вторую задачу на наличие ошибок;

19. Результат не выдал ошибок;

Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def geometric_mean(*args):
6     if args:
7         multi = 1
8         for i in args:
9             multi *= i
10        sr_geom = pow(multi, 1/len(args))
11        return float(sr_geom)
12
13    else:
```

No syntax errors detected :)

Рисунок 8 – Проверка второй задачи на наличие ошибок

- 20. С помощью сайта проверил третью задачу на наличие ошибок;
- 21. Результат не выдал ошибок;

Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def sort(**kwargs):
6     s_num = []
7     [s_num.append(s) for s in kwargs.keys()]
8     values = []
9     [values.append(v) for v in kwargs.values()]
10    values.sort()
11
12    return dict(zip(s_num, values))
13
```

No syntax errors detected :)

Рисунок 9 – Проверка третьей задачи на наличие ошибок

- 22. С помощью сайта проверил индивидуальное задание на наличие ошибок;
- 23. Результат не выдал ошибок;

Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def sum_of_arguments(*args):
6     a = 1
7     negative = 0
8     for idx, n in enumerate(args):
9         if n < 0:
10             args = args[idx+1:]
11             negative += 1
12             break
13     for idx, n in enumerate(args):
```

No syntax errors detected :)

Рисунок 10 – Проверка индивидуального задания на наличие ошибок

24. Отправил все изменения на репозиторий.

Контрольные вопросы

1. Какие аргументы называются позиционными в Python?

При вызове функции аргументы можно передавать как позиционные, которые передаются в том же порядке, в котором они определены при создании функции. То есть, порядок передачи аргументов определяет, какое значение получит каждый аргумент.

2. Какие аргументы называются именованными в Python?

Именованные аргументы – это аргументы передаваемые с именами. При вызове функции можно использовать имена параметров из ее определения. Благодаря `**kwargs` создается словарь, в котором содержатся именованные аргументы, переданные функции при её вызове.

3. Для чего используется оператор `*`?

Оператор «`*`» позволяет «распаковывать» объекты, внутри которых хранятся некие элементы.

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

`*args` – это сокращение от «arguments» (аргументы), а `**kwargs` – сокращение от «keyword arguments» (именованные аргументы). Каждая из этих конструкций используется для распаковки аргументов соответствующего типа, позволяя вызывать функции со списком аргументов переменной длины.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.