

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

Отчет по лабораторной работе № 2.10
Функции с переменным числом параметров в Python

Выполнил студент группы ИВТ-б-о-20-1

Ищенко М.А.

Работа защищена « » _____ 20__ г.

Проверил(а) _____

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

Создан общедоступный репозиторий на GitHub. Дополнен файл .gitignore необходимыми правилами для работы с IDE PyCharm.

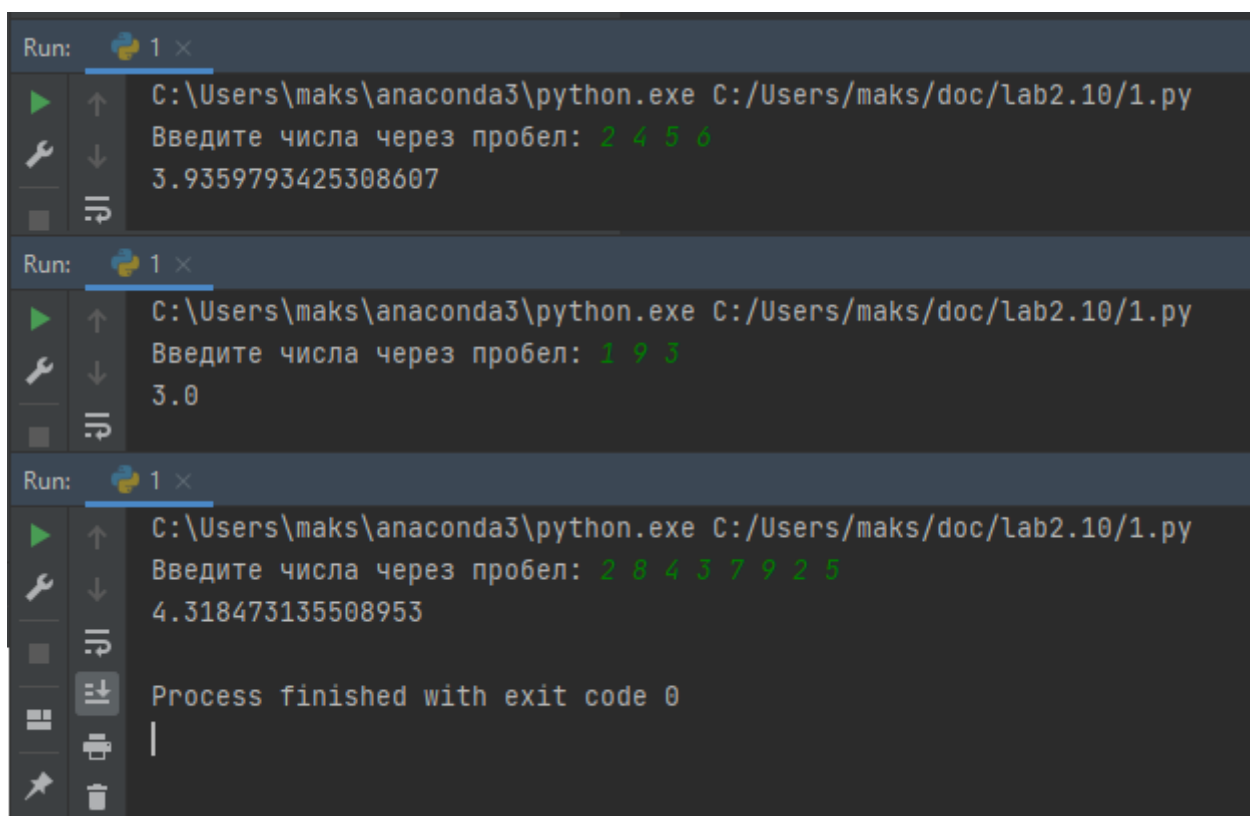
Решён пример из лабораторной, рис. 1



```
Run: primer1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.10/primer1.py
None
6.0
4.5
Process finished with exit code 0
```

Рисунок 1 – Пример

Решены общие задания из методички, рис. 2-4

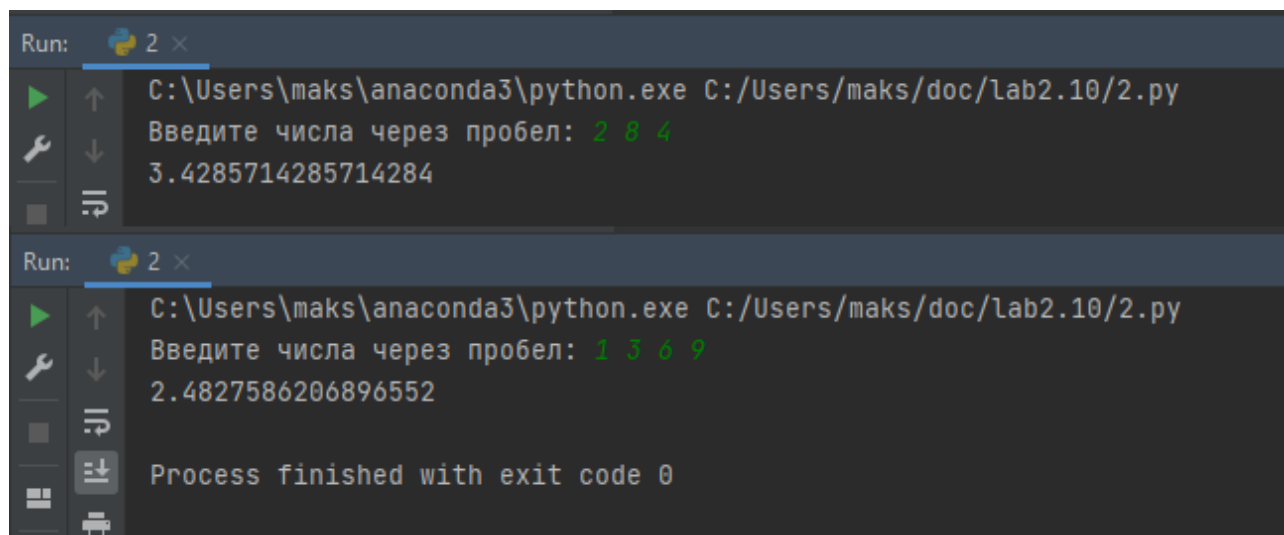


```
Run: 1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.10/1.py
Введите числа через пробел: 2 4 5 6
3.9359793425308607

Run: 1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.10/1.py
Введите числа через пробел: 1 9 3
3.0

Run: 1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.10/1.py
Введите числа через пробел: 2 8 4 3 7 9 2 5
4.318473135508953
Process finished with exit code 0
|
```

Рисунок 2 – Решения задания 1

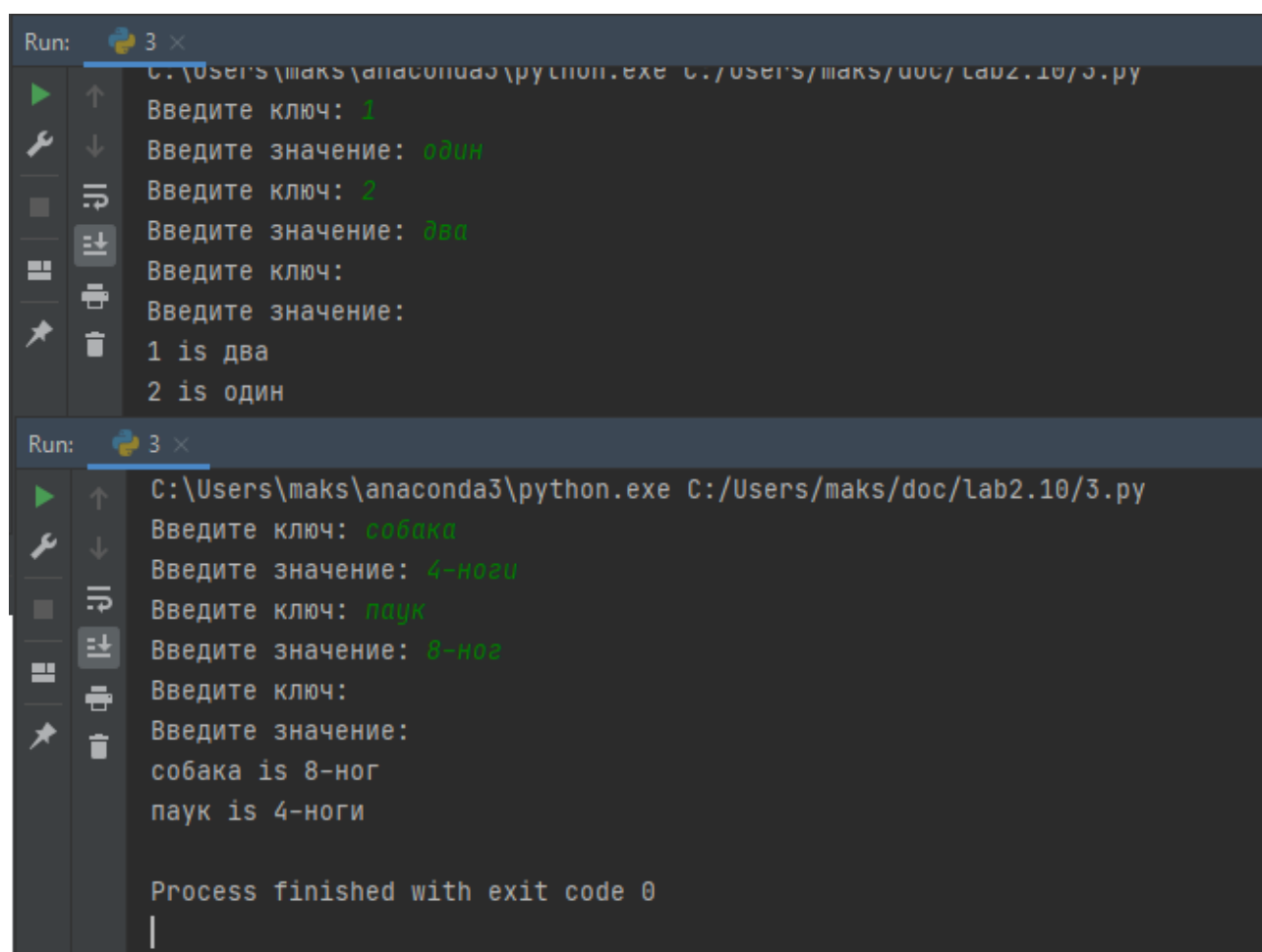


```
Run: 2 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.10/2.py
Введите числа через пробел: 2 8 4
3.4285714285714284

Run: 2 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.10/2.py
Введите числа через пробел: 1 3 6 9
2.4827586206896552

Process finished with exit code 0
```

Рисунок 3 – Решения задания 2



```
Run: 3 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.10/3.py
Введите ключ: 1
Введите значение: один
Введите ключ: 2
Введите значение: два
Введите ключ:
Введите значение:
1 is два
2 is один

Run: 3 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.10/3.py
Введите ключ: собака
Введите значение: 4-ноги
Введите ключ: паук
Введите значение: 8-ног
Введите ключ:
Введите значение:
собака is 8-ног
паук is 4-ноги

Process finished with exit code 0
|
```

Рисунок 4 – Решения задания 3

Выполнено индивидуальное задание варианта 6, рис. 5

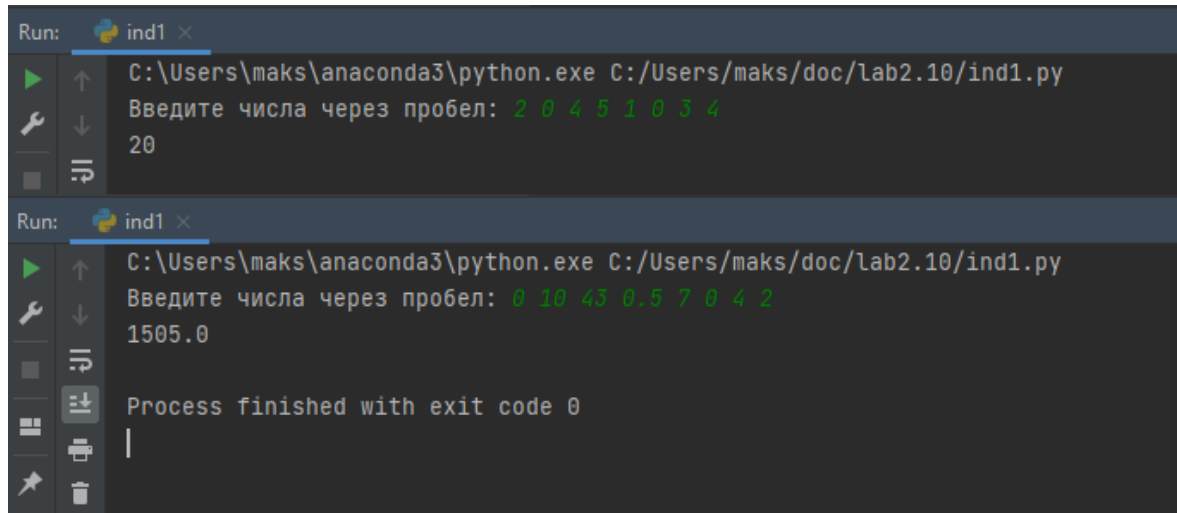


Рисунок 5 – Индивидуальное задание

Программы проверены на flake8, рис. 6

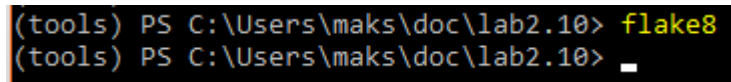


Рисунок 6 – Проверка заданий

Контрольные вопросы:

1. Какие аргументы называются позиционными в Python?

При вызове функции аргументы можно передавать как позиционные – 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?

Аргументы, передаваемые с именами, называются именованными. 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 `*`?

Этот оператор позволяет «распаковывать» объекты, внутри которых хранятся некие элементы.

```
a = [1, 2, 3]
```

```
b = [*a, 4, 5, 6]
```

```
print(b)
```

```
# [1, 2, 3, 4, 5, 6]
```

Тут берётся содержимое списка `a`, распаковывается, и помещается в список `b`.

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 занятия были приобретены навыки по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x