

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ

по лабораторной работе №2.10

Дисциплина: «Программирование на Python»
Тема: «Функции с переменным числом параметров в Python»

Выполнил:
Епифанов Алексей Александрович
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
», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Роман Александрович

(подпись)

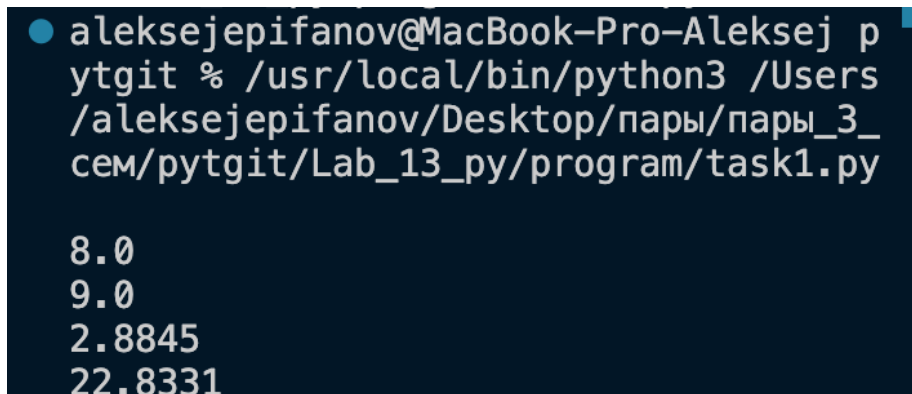
Отчет защищен с оценкой _____ Дата защиты _____
Ставрополь, 2023 г.

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

Порядок выполнения работы:

1. Создал новый репозиторий, клонировал его, в нем создал ветку developer и перешел на нее.
2. Решил задачу 1: Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов. Если функции передается пустой список аргументов, то она должна возвращать значение None .

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
def geometric_mean(*args):
    """
    Поиск среднего геометрического аргументов
    """
    if args:
        values = [float(arg) for arg in args]
        product = 1.0
        for value in values:
            product *= value
        return round(pow(product, 1/len(values)), 4)
    else:
        return None
if __name__ == "__main__":
    print(geometric_mean(4, 8, 16))
    print(geometric_mean(3, 9, 27))
    print(geometric_mean(2, 3, 4))
    print(geometric_mean(31, 12, 32))
```



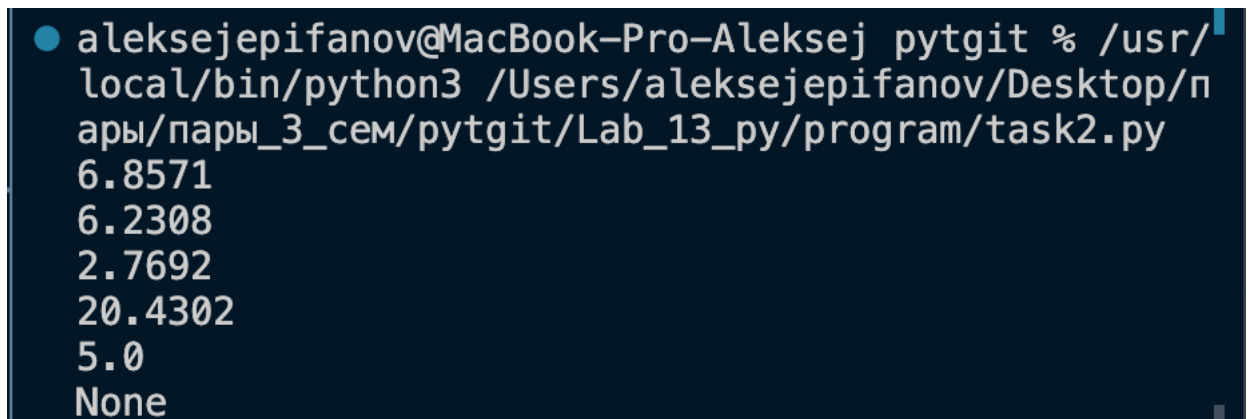
```
● aleksejepifanov@MacBook-Pro-Aleksej p
ytgit % /usr/local/bin/python3 /Users
/aleksejepifanov/Desktop/пары/пары_3_
сем/pytgit/Lab_13_py/program/task1.py

8.0
9.0
2.8845
22.8331
```

Рисунок 1. Вывод программы task1

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

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
def harmonic_mean(*args):
    """
    Поиск среднего гармонического аргументов
    """
    if args:
        values = [float(arg) for arg in args]
        product = 0
        for value in values:
            product += 1/value
        return round(len(values)/product, 4)
    else:
        return None
if __name__ == "__main__":
    print(harmonic_mean(4, 8, 16))
    print(harmonic_mean(3, 9, 27))
    print(harmonic_mean(2, 3, 4))
    print(harmonic_mean(31, 12, 32))
    print(harmonic_mean(5))
    print(harmonic_mean())
```



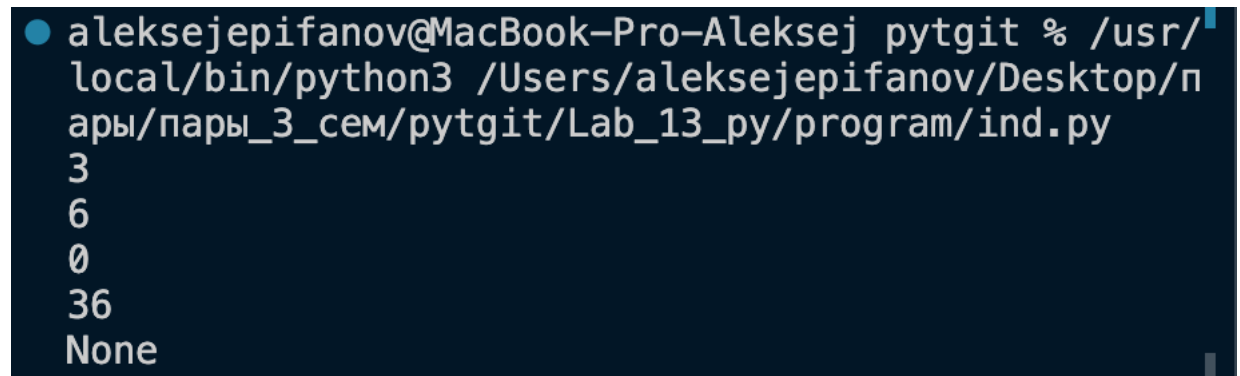
```
aleksejepifanov@MacBook-Pro-Aleksej pytgit % /usr/
local/bin/python3 /Users/aleksejepifanov/Desktop/п
ары/пары_3_сем/pytgit/Lab_13_py/program/task2.py
6.8571
6.2308
2.7692
20.4302
5.0
None
```

Рисунок 2. Вывод программы task2

4. Решил индивидуальное задание вариант 10: Напишите функцию, принимающую произвольное количество аргументов, и возвращающую Сумму модулей аргументов, расположенных после первого отрицательного 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.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
def func(*args):
    """
    Подсчет суммы модулей аргументов после первого отрицательного
    """
    if not args:
        return None
    switch = False
    summ = 0
    for arg in args:
        if not switch and arg < 0:
            switch = True
        elif switch:
            summ += abs(arg)
    return summ
if __name__ == "__main__":
    print(func(1, 2, 1, 2, -1, 3))
    print(func(1, 2, -1, 2, -1, 3))
    print(func(1, -41))
    print(func(1, -41, 34, -2))
    print(func())
```



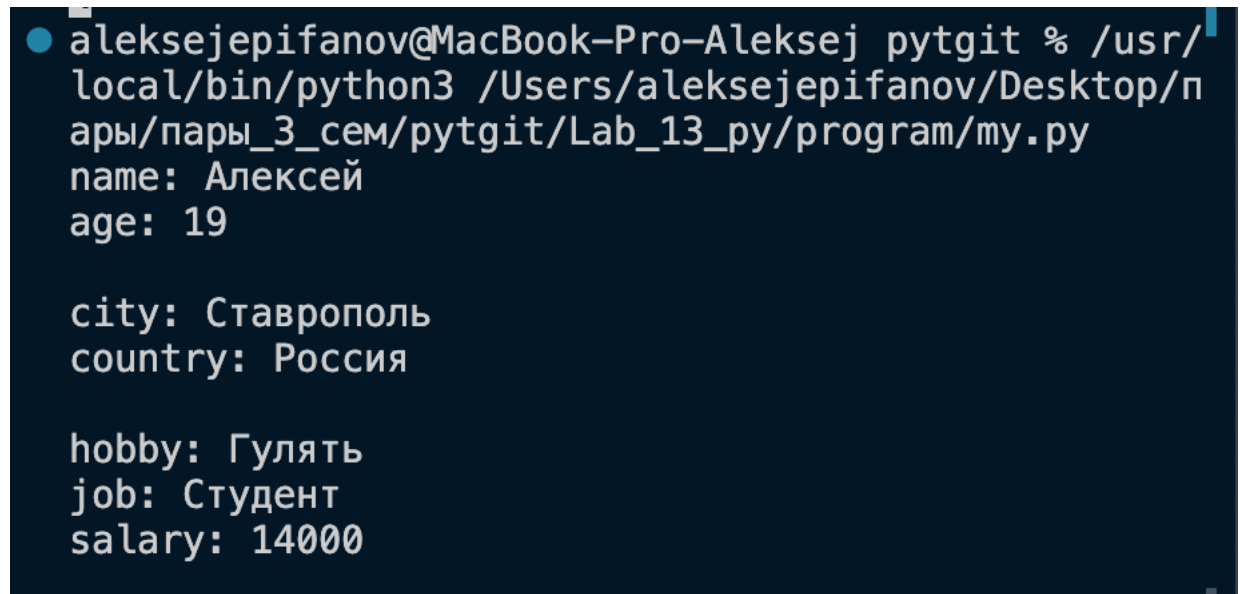
```
aleksejepifanov@MacBook-Pro-Aleksej pytgit % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_3_сем/pytgit/Lab_13_py/program/ind.py
3
6
0
36
None
```

Рисунок 3. Вывод программы ind

5. Выполнил задание: Самостоятельно подберите или придумайте задачу с переменным числом именованных аргументов. Приведите решение этой задачи.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
def my_func(**kwargs):
    """
```

```
Вывод на экран аргументов, переданных в функцию, с их названиями
"""
for key, value in kwargs.items():
    print("{0}: {1}".format(key, value))
print()
if __name__ == "__main__":
    my_func(name="Алексей", age=19)
    my_func(city="Ставрополь", country="Россия")
    my_func(hobby="Гулять", job="Студент", salary=14000)
```



```
aleksejepifanov@MacBook-Pro-Aleksej pytgit % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_3_сем/pytgit/Lab_13_py/program/my.py
name: Алексей
age: 19

city: Ставрополь
country: Россия

hobby: Гулять
job: Студент
salary: 14000
```

Рисунок 4. Вывод программы my

Ответы на контрольные вопросы:

1. Какие аргументы называются позиционными в Python?

В Python аргументы называются позиционными, если они передаются функции в том же порядке, в котором они определены в функции.

В функцию также можно передать переменное количество позиционных аргументов. Это делается с помощью оператора * перед именем аргумента в определении функции.

2. Какие аргументы называются именованными в Python?

В Python аргументы называются именованными, если они передаются функции с указанием имени аргумента, за которым следует значение аргумента.

В функцию также можно передать переменное количество именованных аргументов. Это делается с помощью оператора `**` перед именем аргумента в определении функции.

3. Для чего используется оператор `*` ?

Оператор `*` чаще всего ассоциируется у людей с операцией умножения, но в Python он имеет и другой смысл. Этот оператор позволяет «распаковывать» объекты, внутри которых хранятся некие элементы.

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

Итак, мы знаем о том, что оператор «звёздочка» в Python способен «вытаскивать» из объектов составляющие их элементы. Знаем мы и о том, что существует два вида параметров функций. А именно, `*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 результате выполнения работы были приобретены навыки по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.