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 №2.10**

# дисциплины «Программирование на Python»

|  |
| --- |
| Выполнил:  Кенесбаев Хилол Куат улы  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 Р.А., доцент кафедры инфокоммуникаций  \_\_\_ \_ \_ \_ \_ \_ \_ \_ \_ \_\_  (подпись) |

Отчет защищен с оценкой \_ \_ Дата защиты\_\_ \_

Ставрополь, 2023 г.

**Тема:** Функции с переменным числом параметров в Python

**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: написать функцию, вычисляющую среднее геометрическое своих аргументов. Если функции передается пустой список аргументов, то она должна возвращать значение None.

# Код программы:

#!/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))

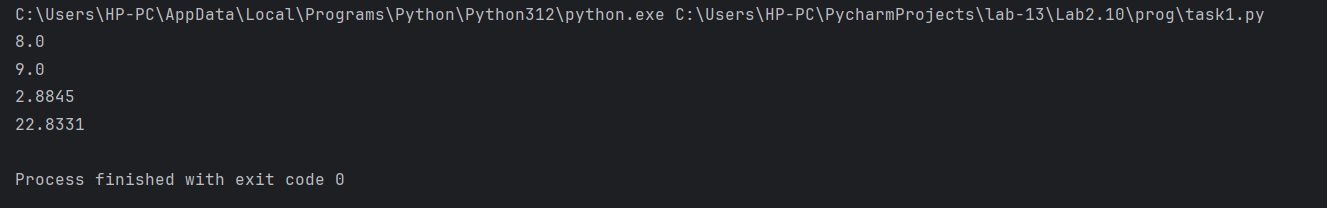


Рисунок 1 Вывод программы task1

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

# Код пограммы:

#!/usr/bin/env python3 # -\*- coding:

utf-8 -\*- defharmonic\_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())

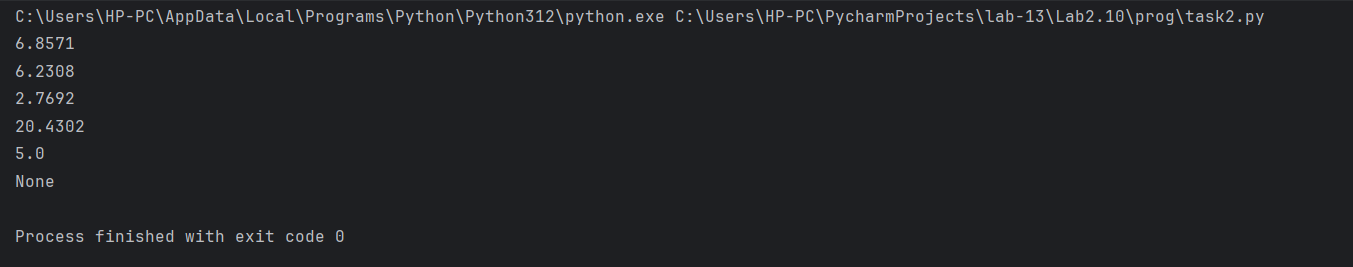


Рисунок 2 Вывод программы task2

1. Решил индивидуальное задание вариант 10: Напишите функцию, принимающую произвольное количество аргументов, и возвращающую сумму модулей, 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.

# Код программы:

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())

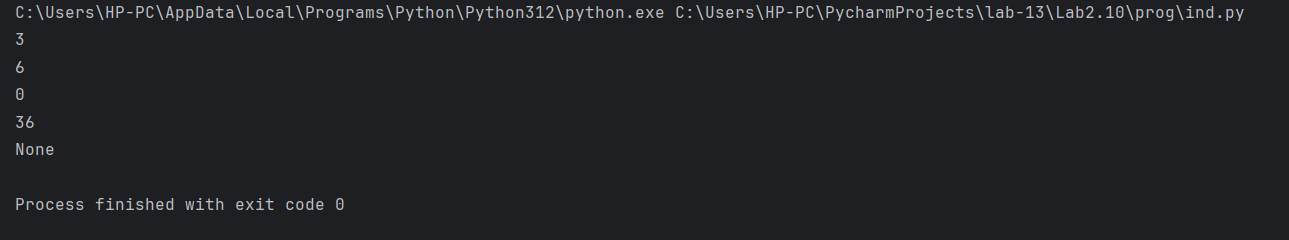


Рисунок 3 Вывод программы ind

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

# Код программы:

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="Хилол", age=21)

my\_func(city="Ставрополь", country="Россия")

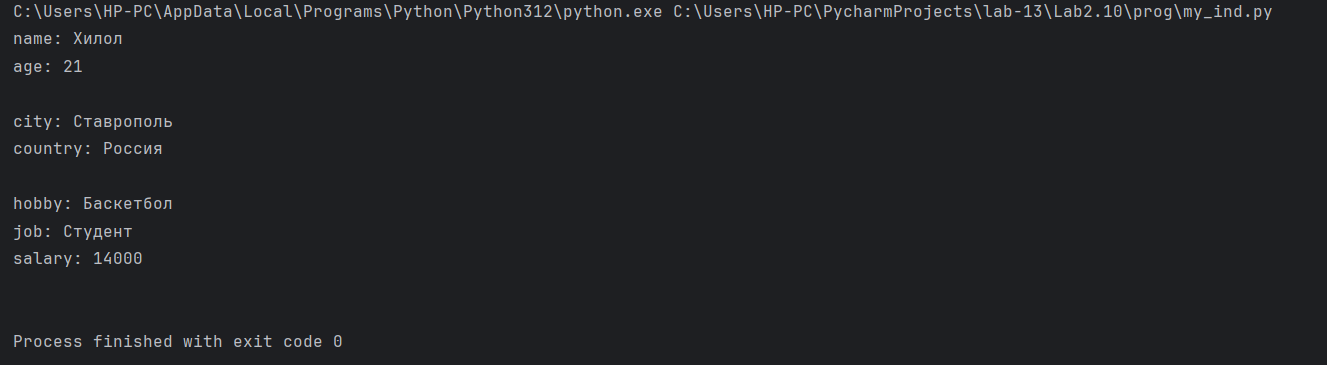
my\_func(hobby="Баскетбол", job="Студент", salary=14000) 

Рисунок 4. Вывод программы my\_ind

# 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.

1. 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.

1. 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.

1. 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.