# МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙ- СКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное образовательное учреждение высшего образования**

# «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**Кафедра инфокоммуникаций**

# Основы кроссплатформенного программирования Отчет по лабораторной работе №2.10

Тема: «Функции с переменным числом параметров в Python»

|  |
| --- |
| Выполнил студент группы  ИВТ-б-о-21-1 |
| Артемьев А.В. « » 20 г. |
| Подпись студента |
| Работа защищена « » 20 г. |
| Проверил доцент  Кафедры инфокоммуникаций, старший преподаватель  Воронкин Р.А.  (подпись) |

Ставрополь 2022

**Цель работы:** приобретение навыков по работе с функциями с перемен- ным числом параметров при написании программ с помощью языка програм- мирования Python версии 3.x.

# Ход работы:

# 

1. **Создал репозиторий в GitHub,** дополнил правила в .gitignore для ра- боты с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на компьютер и организовал в соответствии с моделью ветвления git-flow.

Рисунок 1.1 – Созданный репозиторий

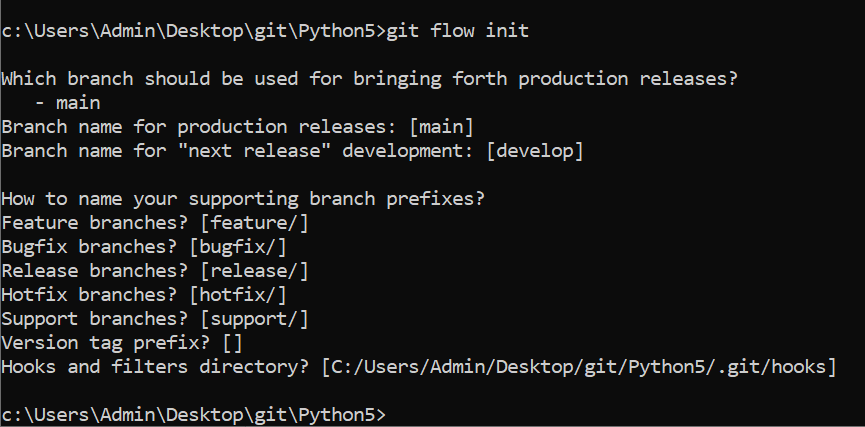


Рисунок 1.2 – Организация репозитория в соответствии с моделью ветвления git-flow

1. **Создал проект** Pycharm в папке репозитория, проработал примеры ЛР.

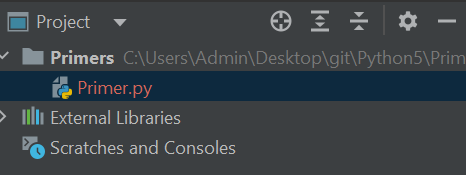


Рисунок 2.1 – Созданные проекты

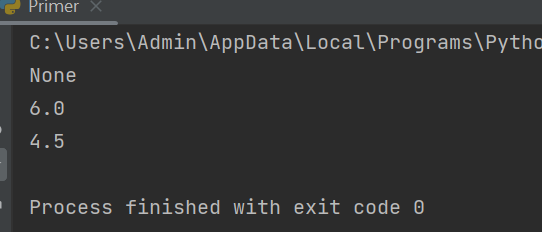


Рисунок 2.2 – Результат работы примера №1

**Задание №1.** Решить поставленную задачу: написать функцию, вычис- ляющую среднее геометрическое своих аргументов a1, a2…an. Если функции передается пустой список аргументов, то она должна возвращать значение None.

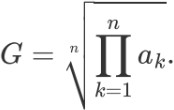


Рисунок 3.1 – Формула для задания №1

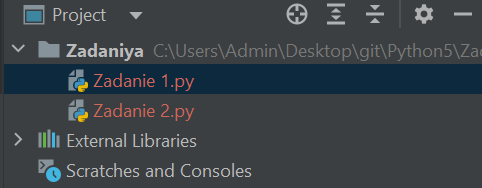


Рисунок 3.2 – Созданные проекты

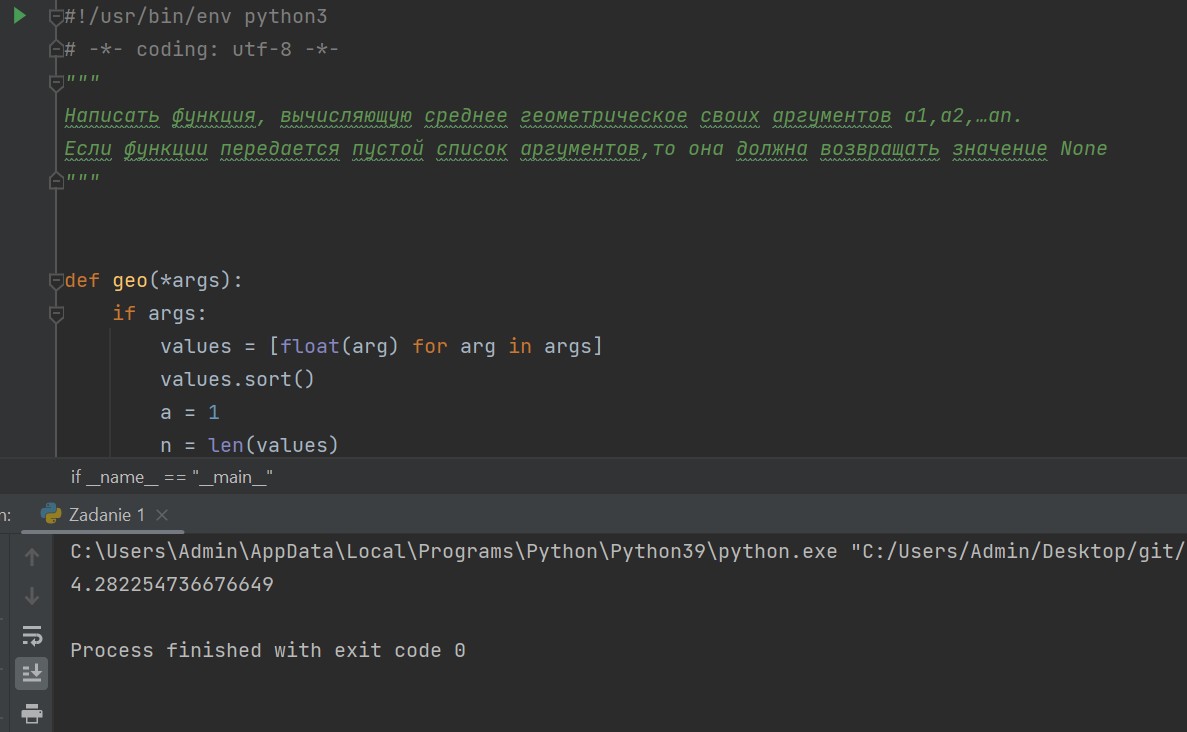


Рисунок 3.3 – Результат работы программы

**Задание №2.** Решить поставленную задачу: написать функцию, вычис- ляющую среднее гармоническое своих аргументов a1, a2…an. Если функции передается пустой список аргументов, то она должна возвращать значение None.

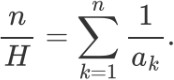


Рисунок 3.4 – Формула для задания №2

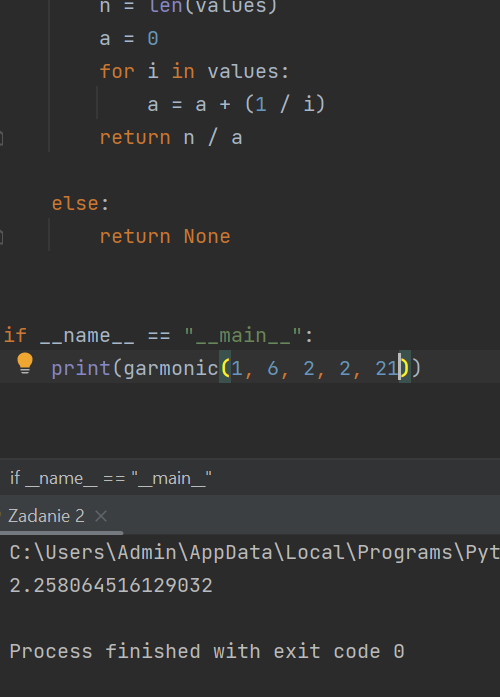


Рисунок 3.5 – Результат работы программы

# №3. Самостоятельно подберите или придумайте задачу с перемен- ным числом именованных аргументов. Приведите решение этой задачи.

Дано время забега на дистанцию 100 метров студентами в виде ключ- значения. Определить лучшее время и среднее время среди всех студентов.

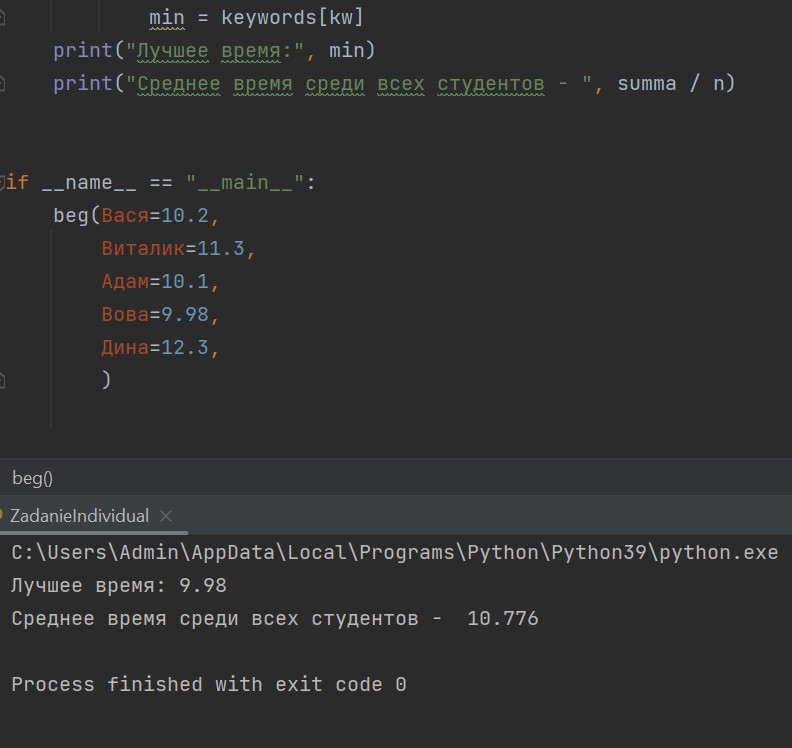


Рисунок 3.6 – Результат выполнения программы

**Индивидуальное задание. В – 1.** Напишите функцию, принимающую произвольное количество аргументов, и возвращающую требуемое значение. Если функции передается пустой список аргументов, то она должна возвра- щать значение None. В процессе решения не использовать преобразования конструкции \*args в список или иную структуру данных.

# Задание: 1. Произведение аргументов, расположенных между мак- симальным и минимальным аргументами.

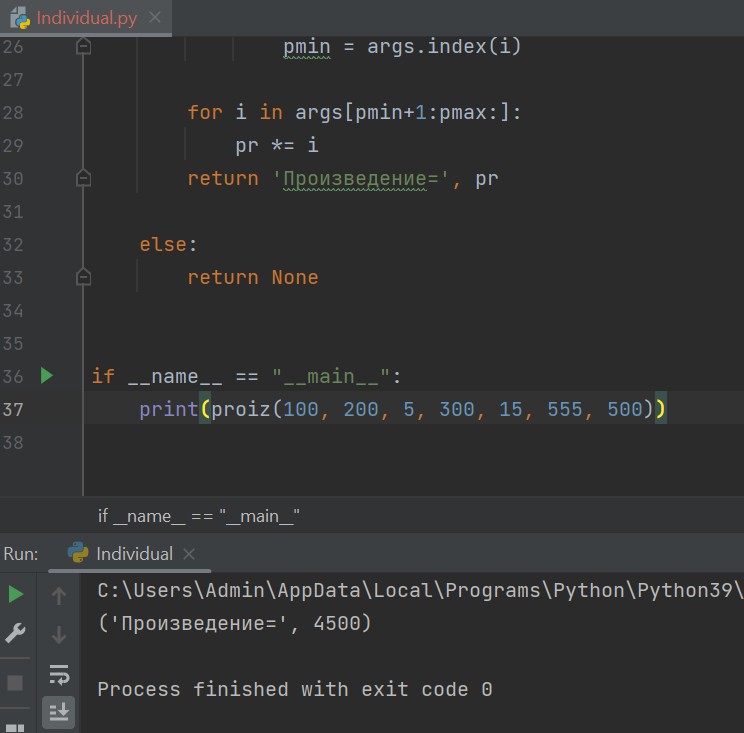


Рисунок 4.1 – Результат выполнения программы

**Вывод:** в результате выполнения лабораторной работы были приобре- тены практические навыки и теоретические сведения по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.

# Ответы на контрольные вопросы:

* 1. **Какие аргументы называются позиционными в Python?**

Аргументы, которые передаются без указания имен называются позици- онными, потому что именно по позиции, расположению аргумента, функция понимает, какому параметру он соответствует.

# Какие аргументы называются именованными в Python?

Аргументы, передаваемые с именами, называются именованными. При вызове функции можно использовать имена параметров из ее определения.

# 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.

Вот пример:

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 спи-

сок b.

# Каково назначение конструкций \*args и \*\*kwargs?

Итак, мы знаем о том, что оператор «звёздочка» в Python способен «вы- таскивать» из объектов составляющие их элементы. Знаем мы и о том, что су- ществует 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.

**Важно помнить, что «args» — это всего лишь набор символов**, кото- рым принято обозначать аргументы. Самое главное тут — это оператор \*. А то, что именно идёт после него, особой роли не играет. Благодаря использова- нию \* мы создали список позиционных аргументов на основе того, что было передано функции при вызове.

После того, как мы разобрались с \*args, с пониманием \*\*kwargs проблем быть уже не должно.

Имя, опять же, значения не имеет. Главное — это два символа \*\*. Бла- годаря им создаётся словарь, в котором содержатся именованные аргументы, переданные функции при её вызове.