

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

**Кафедра инфокоммуникаций
«Функции с переменным числом параметров в Python»**

**Отчет по лабораторной работе № 2.10
по дисциплине «Программирование на Python»**

Выполнил студент группы ИВТ-б-о-21-1

Шайдеров Дмитрий Викторович.

«24 » октября 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

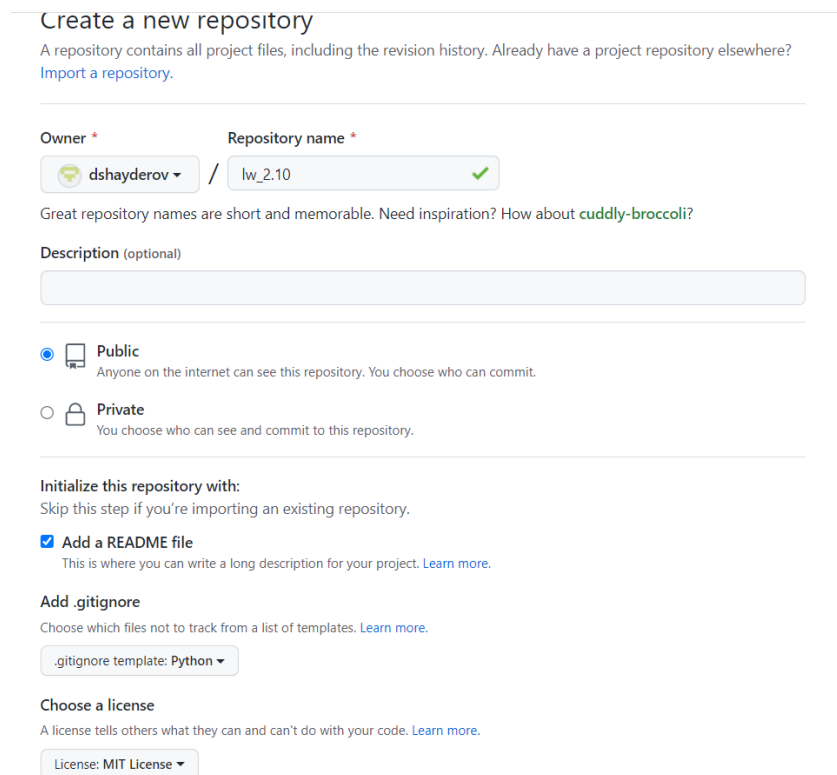
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

Цель работы: 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 общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * Repository name *

dshayderov / lw_2.10 ✓

Great repository names are short and memorable. Need inspiration? How about [cuddly-broccoli?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

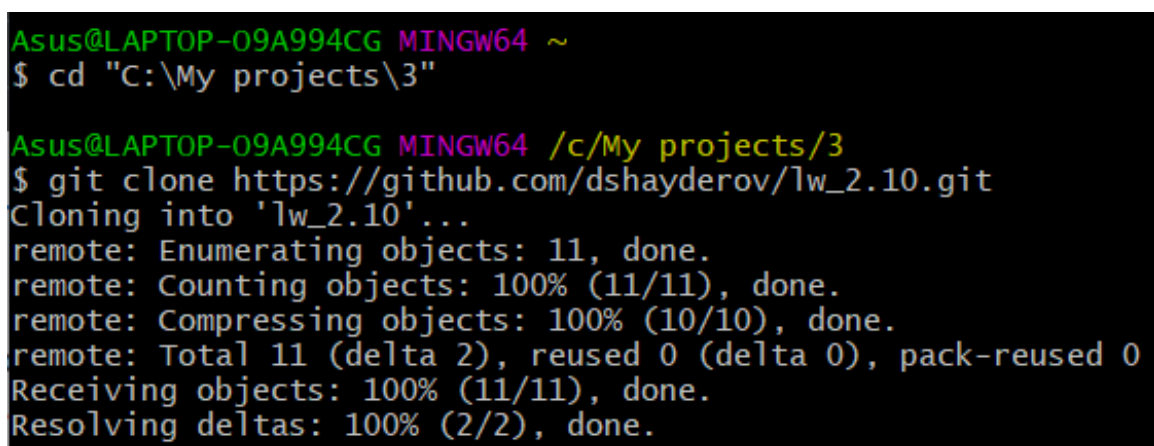
.gitignore template: Python

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.



```
Asus@LAPTOP-09A994CG MINGW64 ~  
$ cd "C:\My projects\3"  
  
Asus@LAPTOP-09A994CG MINGW64 /c/My projects/3  
$ git clone https://github.com/dshayderov/lw_2.10.git  
Cloning into 'lw_2.10'...  
remote: Enumerating objects: 11, done.  
remote: Counting objects: 100% (11/11), done.  
remote: Compressing objects: 100% (10/10), done.  
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (11/11), done.  
Resolving deltas: 100% (2/2), done.
```

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```
Asus@LAPTOP-09A994CG MINGW64 ~  
$ cd "C:\My projects\3\lw_2.10"  
  
Asus@LAPTOP-09A994CG MINGW64 /c/My projects/3/lw_2.10 (main)  
$ git checkout -b develop  
Switched to a new branch 'develop'  
  
Asus@LAPTOP-09A994CG MINGW64 /c/My projects/3/lw_2.10 (develop)  
$ |
```

Рисунок 3 - Ветвление по модели git-flow

4. Создайте проект PyCharm в папке репозитория.

Этот компьютер > OS (C:) > My projects > 3 > lw_2.10 >				
Имя	Дата изменения	Тип	Размер	
Project	24.10.2022 9:47	Папка с файлами		
.gitignore	18.10.2022 12:45	Файл "GITIGNORE"	2 КБ	
LICENSE	18.10.2022 12:45	Файл	2 КБ	
README.md	18.10.2022 12:45	Файл "MD"	1 КБ	

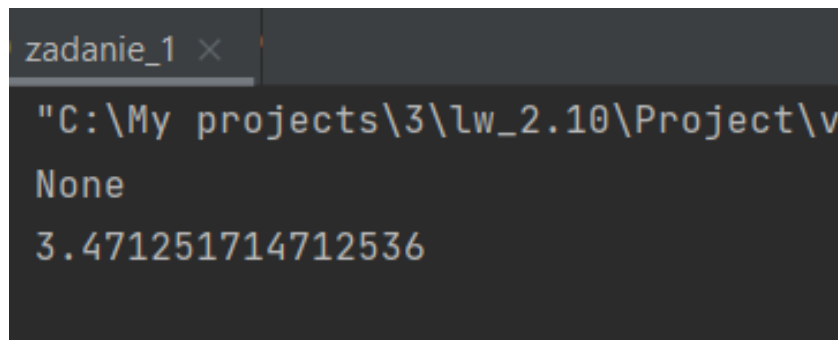
Рисунок 4 - Проект PyCharm

5. Проработайте пример лабораторной работы. Создайте для него отдельный модуль языка Python. Зафиксируйте изменения в репозитории.

```
primer x  
"C:\My projects\3\lw_2.10\Project\ve  
None  
6.0  
4.5
```

Рисунок 5 - Результат выполнения примера

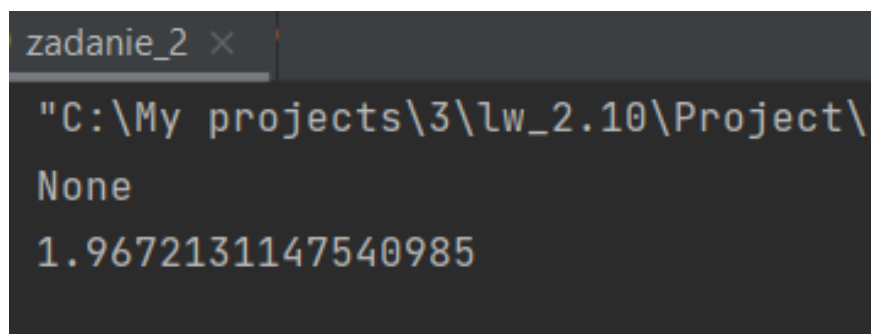
6. Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов a_1, a_2, \dots, a_n .



```
zadanie_1 ×  
"C:\My projects\3\lw_2.10\Project\w  
None  
3.471251714712536
```

Рисунок 6 - Результат выполнения задания 1

7. Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов a_1, a_2, \dots, a_n .

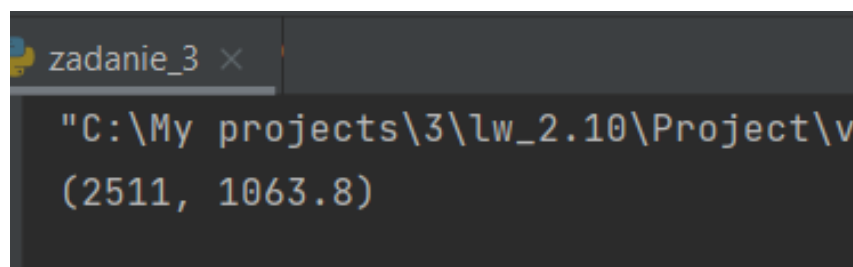


```
zadanie_2 ×  
"C:\My projects\3\lw_2.10\Project\w  
None  
1.9672131147540985
```

Рисунок 7 - Результат выполнения задания 2

8. Самостоятельно подберите или придумайте задачу с переменным числом именованных аргументов. Приведите решение этой задачи.

Даны значения площади городов в виде ключ-значения. Определить наибольшую и среднюю площадь.

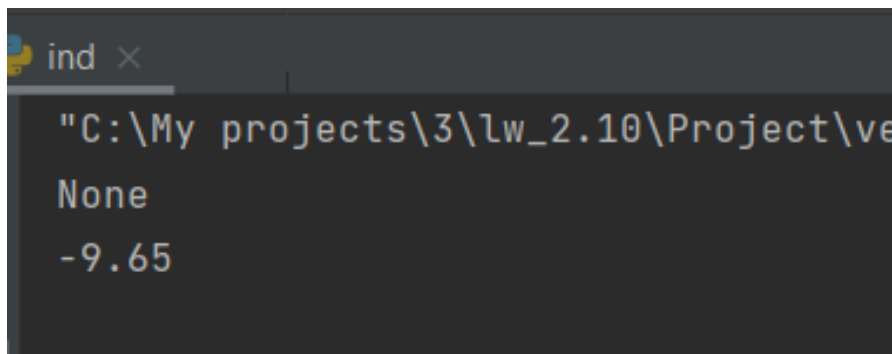


```
zadanie_3 ×  
"C:\My projects\3\lw_2.10\Project\w  
(2511, 1063.8)
```

Рисунок 8 - Результат выполнения задания 3

9. Решите индивидуальное задание. (Вариант 26/8)

Сумму аргументов, расположенных между первым и вторым положительными аргументами.



```
ind x
"C:\My projects\3\lw_2.10\Project\ve
None
-9.65
```

Рисунок 9 - Результат выполнения индивидуального задания

Контрольные вопросы:

1. Какие аргументы называются позиционными в Python?

В программировании рекурсия — вызов функции (процедуры) из неё же самой, непосредственно (простая рекурсия) или через другие функции (сложная или косвенная рекурсия). Рекурсивная программа позволяет описать повторяющееся или даже потенциально бесконечное вычисление, причём без явных повторений частей программы и использования циклов.

2. Какие аргументы называются именованными в Python?

База рекурсии – это такие аргументы функции, которые делают задачу настолько простой, что решение не требует дальнейших вложенных вызовов.

3. Для чего используется оператор * ?

Стек — это структура данных, в которой элементы хранятся в порядке поступления.

Стек хранит последовательность данных. Связаны данные так: каждый элемент указывает на тот, который нужно использовать следующим. Это линейная связь — данные идут друг за другом и нужно брать их по очереди. Из середины стека брать нельзя.

Главный принцип работы стека — данные, которые попали в стек недавно, используются первыми. Чем раньше попал — тем позже используется. После использования элемент стека исчезает, и верхним становится следующий элемент.

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

Функция `sys.getrecursionlimit()` возвращает текущее значение предела рекурсии, максимальную глубину стека интерпретатора Python. Этот предел предотвращает бесконечную рекурсию от переполнения стека языка C и сбоя Python. Это значение может быть установлено с помощью `sys`.

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