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

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.10

Дисциплина: «Функции с переменным числом параметров в Python»

Тема: «Рекурсия в языке Python»

Выполнил: студент 1 курса

группы ИВТ-б-о-21-1

Кочкаров Умар Ахматович

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.

Ход работы

1. Создать общедоступный репозиторий с лицензией MIT и языком Python.

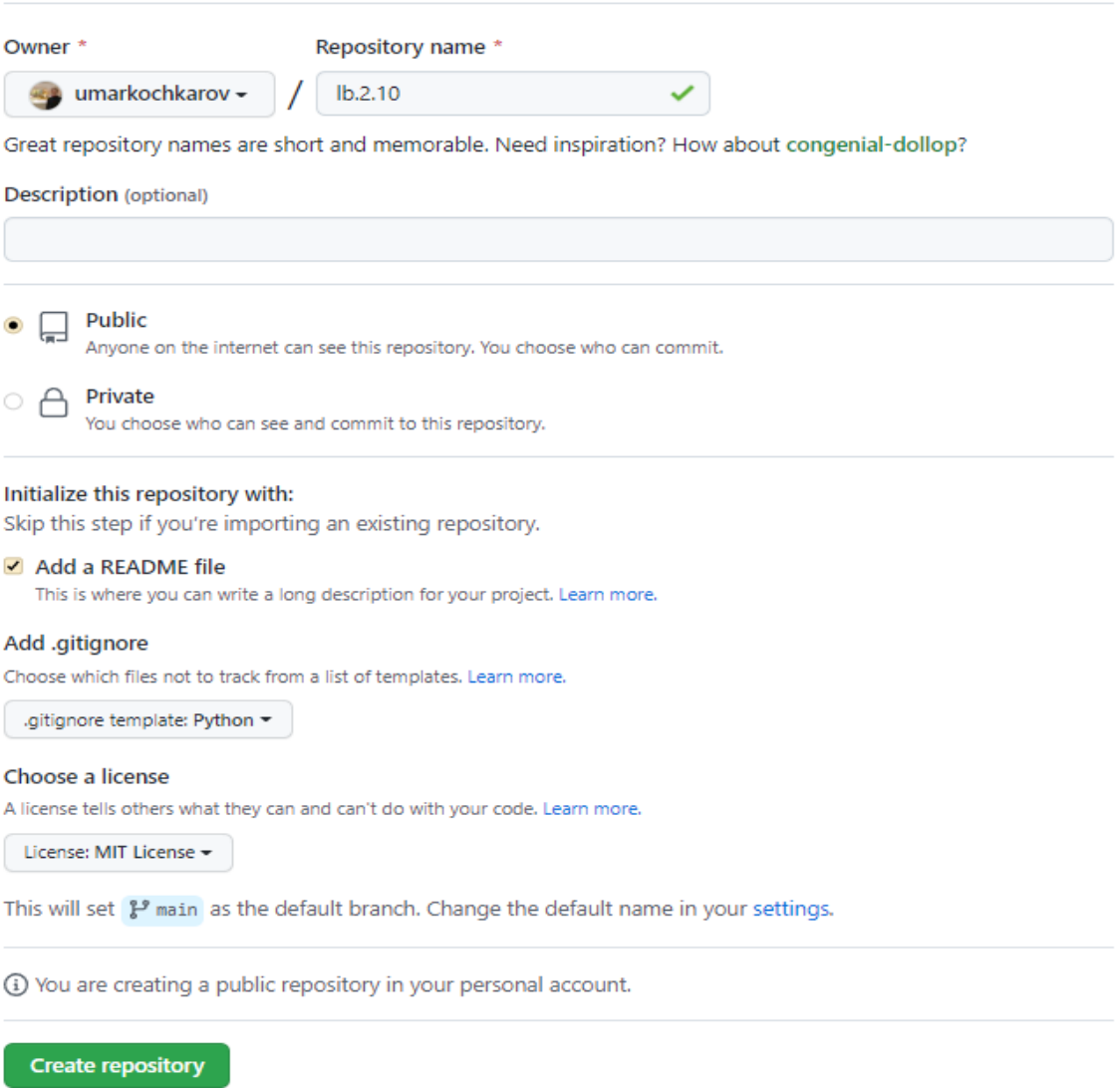

Рисунок 1. Создание репозитория

2. Клонировать репозиторий на ПК:

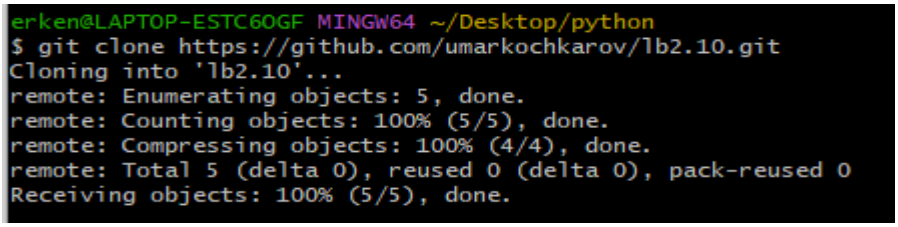

Рисунок 2. Клонирование

3. Организовать репозиторий в соответствии с моделью ветвления git-flow.

```
erken@LAPTOP-ESTC6OGF MINGW64 ~/Desktop/python/Лаб2.10/lb2.10 (main)
$ git flow init

Which branch should be used for bringing forth production releases?
   - main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?

Feature branches? [feature/] Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/erken/Desktop/python/Лаб2.10/lb2.10/.git/
hooks]
```

Рисунок 3. Организация репозитория в соответствии с git-flow

4. Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов

$$G = \sqrt[n]{\prod_{k=1}^{n} a_k}.$$

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import math

def Geom(*args):
    if args:
        values = [float(arg) for arg in args]
        values.sort()

        a = 1
        n = len(values)
        for i in values:
            a = a * pow(i, (1/n))
        return a
    else:
        return None

if __name__ == "__main__":
    print(Geom())
    print(Geom(3,4,5))
```

```
Run - 2.10
Run:  1
C:\Users\erken\Desktop\python\Примеры\2.10\venv\Scripts\
None
3.9148676411688625
```

Рисунок 4. Результат выполнения программы

5. Решить поставленную задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import math

def Geom(*args):
    if args:
        values = [float(arg) for arg in args]
        values.sort()

        a = 1
        n = len(values)
        for i in values:
            a = a * pow(i, (1/n))
        return a
    else:
        return None

if __name__ == "__main__":
    print(Geom())
    print(Geom(3,4,5))
```

```
Run - 2.10
Run:  2
C:\Users\erken\Desktop\python\Примеры\2.10\venv\Scripts
None
2.18978102189781
```

Рисунок 5. Результат выполнения программы

6. Индивидуальное задание(вариант 8)

Сумму аргументов, расположенных между первым и вторым положительными аргументами.

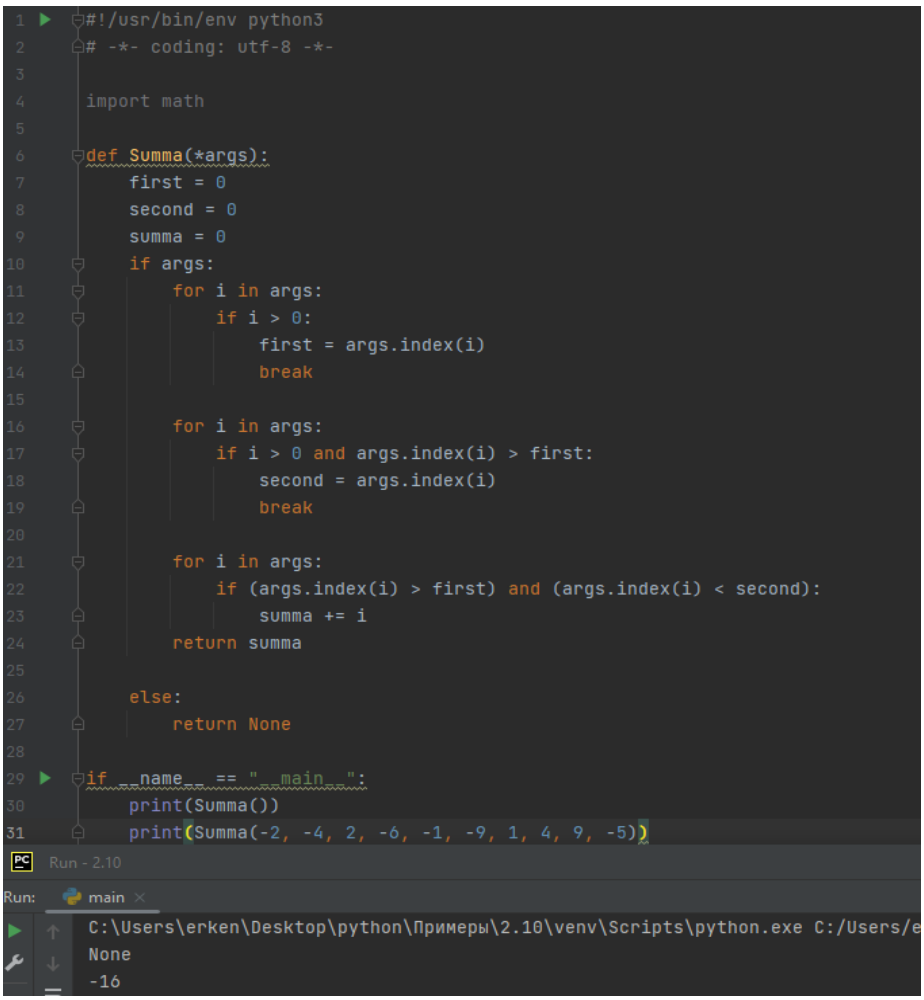

Рисунок 6. Индивидуальное задание

Ответы на вопросы:

1. Какие аргументы называются позиционными в Python?

Это аргументы, передаваемые в вызов в определенной последовательности (на определенных позициях), без указания их имен.

Элементы объектов, поддерживающих итерирование, могут использоваться в качестве позиционных аргументов, если их распаковывать при помощи *.

2. Какие аргументы называются именованными в Python?

Эти аргументы, передаваемые в вызов при помощи имени (идентификатора), либо словаря с его распаковкой при помощи **.

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 число параметров при написании программ с помощью языка программирования Python версии 3.x