

**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 «Основы программной инженерии»**

Выполнил студент группы ПИЖ-б-о-21-1

Кучеренко С. Ю. « » 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

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

Выполнение работы:

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия IT и язык программирования Python.

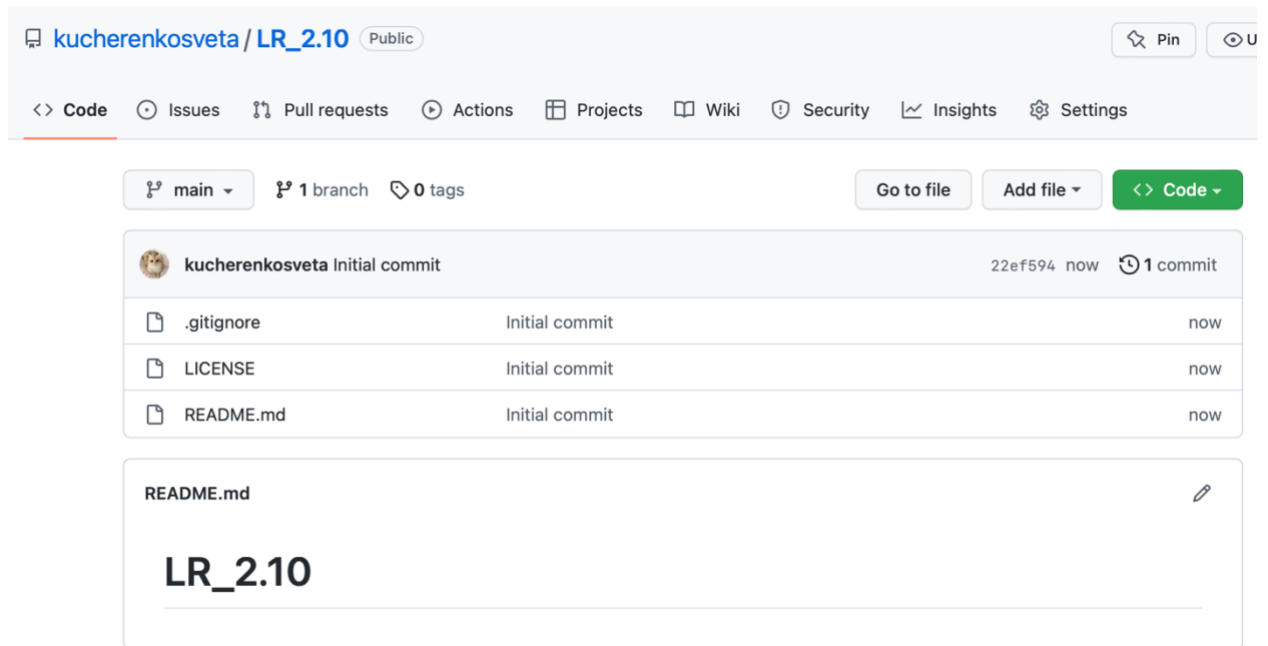


Рисунок 1 – Создание репозитория

3. Выполните клонирование созданного репозитория.

```
[(base) svetik@MacBook-Air-Svetik Laba13 % git clone https://github.com/kucherenkosveta/LR_2.10.git
Cloning into 'LR_2.10'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2 – Клонирование репозитория

4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.

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

```
[(base) svetik@MacBook-Air-Svetik LR_2.10 % 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? [/Users/svetik/Desktop/Laba13/LR_2.10/.git/hooks]
(base) svetik@MacBook-Air-Svetik LR_2.10 %
```

Рисунок 3 – Организация репозитория в соответствии с моделью git-flow

6. Создайте проект PyCharm в папке репозитория.

7. Проработайте пример лабораторной работы.

Пример 1. Разработать функцию для определения медианы значений аргументов функции. Если функции передается пустой список аргументов, то она должна возвращать значение None.

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

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

        n = len(values)
        idx = n // 2
        if n % 2:
            return values[idx]
        else:
            return (values[idx - 1] + values[idx]) / 2
    else:
        return None

if __name__ == "__main__":
    print(median())
    print(median(3, 7, 1, 6, 9))
    print(median(1, 5, 8, 4, 3, 9))
```

```
None
6.0
4.5
```

Рисунок 4 – Результат работы программы

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

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

Если функции передается пустой список аргументов, то она должна возвращать значение None.

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

def sr_geom(*args):
    if args:
        values = [float(arg) for arg in args]
        g = 1
        for arg in values:
            g = g * arg
        n = len(args)
        return pow(g, 1/n)
    else:
        return None

if __name__ == "__main__":
    print(f'Среднее геометрическое 1: {sr_geom(1, 2, 3, 4, 5)}')
    print(f'Среднее геометрическое 1: {sr_geom()}')
    print(f'Среднее геометрическое 1: {sr_geom(1.3, 7.9, 8.3)}')
```

```
Среднее геометрическое 1: 2.605171084697352
Среднее геометрическое 1: None
Среднее геометрическое 1: 4.400981186140995
```

Рисунок 5 – Результат работы программы

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

$$\frac{n}{H} = \sum_{k=1}^n \frac{1}{a_k}.$$

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

def sr_harm(*args):
    if args:
        values = [float(arg) for arg in args]
        h = 0
        for arg in values:
            h += 1 / arg
        n = len(args)
        return n / h
    else:
        return None

if __name__ == "__main__":
    print(f'Среднее гармоническое: {sr_harm(1, 2, 3, 4, 5, 6)}')
    print(f'Среднее гармоническое: {sr_harm()}')
    print(f'Среднее гармоническое: {sr_harm(3.3, 6.5, 9.7, 8.0)}')
```

```
Среднее гармоническое: 2.4489795918367347
Среднее гармоническое: None
Среднее гармоническое: 5.83967828653373
```

Рисунок 6 – Результат работы программы

10. Зафиксируйте сделанные изменения в репозитории.

11. Решите индивидуальное задание согласно своему варианту.

Напишите функцию, принимающую произвольное количество аргументов, и возвращающую требуемое значение. Если функции передается пустой список аргументов, то она должна возвращать значение None. Номер варианта определяется по согласованию с преподавателем.

В процессе решения не использовать преобразования конструкции `*args` в список или иную структуру данных.

(Вариант 9) Сумму модулей аргументов, расположенных после первого аргумента, равного нулю.

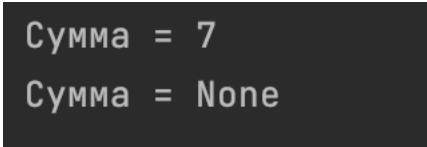
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

# Сумма модулей аргументов, расположенных после первого аргумента, равного
# нулю

def summa(*args):
    if args:
        return sum(map(abs, args[args.index(0) + 1:]))

    else:
        return None

if __name__ == '__main__':
    print(f'Сумма = {summa(7, -3, 8, 0, -1, 5, 1)}')
    print(f'Сумма = {summa()}')
```



```
Сумма = 7
Сумма = None
```

Рисунок 7 – Результат работы программы

12. Зафиксируйте сделанные изменения в репозитории.

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

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

def print_harry_potter(**kwargs):
    if kwargs:
        for movie_number, movie_title in kwargs.items():
            print(f'{movie_number}: {movie_title}')
    else:
        return None

if __name__ == "__main__":
    print_harry_potter(
        One="Harry Potter and the Philosopher's Stone",
        Two="Harry Potter And The Chamber of secrets",
        Three="Harry Potter and the prisoner of Azkaban",
        Four="Harry Potter and the Goblet of Fire",
        Five="Harry Potter and the Order of the Phoenix",
        Six="Harry Potter and the Half-Blood Prince",
        Seven="Harry Potter and the Deathly Hallows. Part 1",
        Eight="Harry Potter and the Deathly Hallows. Part 2"
    )
```

```
One: Harry Potter and the Philosopher's Stone
Two: Harry Potter And The Chamber of secrets
Three: Harry Potter and the prisoner of Azkaban
Four: Harry Potter and the Goblet of Fire
Five: Harry Potter and the Order of the Phoenix
Six: Harry Potter and the Half-Blood Prince
Seven: Harry Potter and the Deathly Hallows. Part 1
Eight: Harry Potter and the Deathly Hallows. Part 2
```

Рисунок 8 – Результат работы программы

12. Зафиксируйте сделанные изменения в репозитории.

Вопросы для защиты работы

1. Какие аргументы называются позиционными в Python?

Это аргументы, передаваемые в вызов в определённой последовательности (на определённых позициях), без указания их имён. Элементы объектов, поддерживающих итерирование, могут использоваться в качестве позиционных аргументов, если их распаковать при помощи *.

2. Какие аргументы называются именованными в Python?

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

3. Для чего используется оператор *?

Этот оператор позволяет «распаковывать» объекты, внутри которых хранятся некие элементы.

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

*args используется для передачи произвольного числа именованных аргументов функции.

****kwargs** позволяет передавать произвольное число именованных аргументов в функцию.