

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: «Программирование на Python»

Тема: «Функции с переменным числом параметров в Python»

Вариант 25

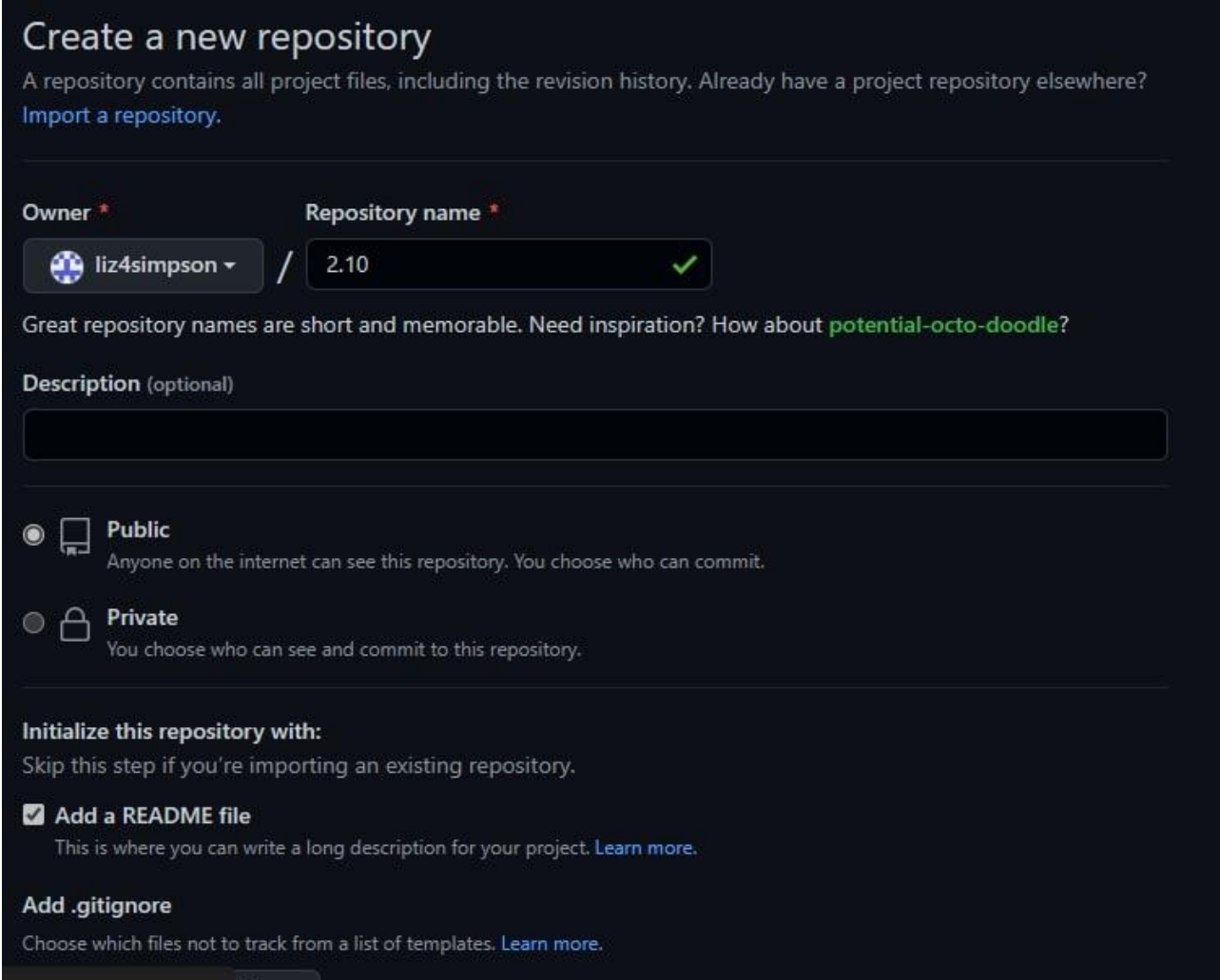
Выполнила: студентка 2
курса группы ИВТ-б-о-21-1
Яковлева Елизавета
Андреевна

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.

Практическая часть:

1. Создала общедоступный репозиторий на GitHub.



Create a new repository

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

Owner **liz4simpson** / Repository name **2.10** ✓

Great repository names are short and memorable. Need inspiration? How about **potential-octo-doodle?**

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.](#)

Рисунок 1. Создание репозитория

2. Выполнила клонирование созданного репозитория.

```
C:\Users\Elizaveta\Desktop\git>git clone https://github.com/liz4simpson/2.10.git
Cloning into '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), 5.17 KiB | 441.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.
```

Рисунок 2. Клонирование репозитория

3. Дополнила файл .gitignore.

```
*.gitignore - Блокнот
Файл  Правка  Формат  Вид  Справка
# Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
# Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm

### PyCharm ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm and Rider
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839
.idea/
.idea

# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf

# AWS User-specific
.idea/**/aws.xml

# Generated files
.idea/**/contentModel.xml
```

Рисунок 3. Изменение файла .gitignore

4. Организовала репозиторий в соответствии git-flow.

```
C:\Users\Elizaveta\Desktop\git\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? [C:/Users/Elizaveta/Desktop/git/2.10/.git/hooks]
```

Рисунок 4. Организация репозитория в соответствии с git-flow

5. Проработала примеры лабораторной работы.

```
#!/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
```

primer ×

```
C:\nana\2.10\Scripts\python.exe C:\Users\Elizaveta\Desktop\git\2.10\pro:
None
6.0
4.5
```

Рисунок 5. Результат выполнения примера

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

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

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

```
C:\nana\2.10\Scripts\python.exe C:\U
None
2.213363839400643
4.035654308729804

Process finished with exit code 0
```

Рисунок 6. Результат выполнения задания 1

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

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

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

```
C:\nana\2.10\Scripts\python.exe C:\Users\
None
1.9200000000000004
2.97111416781293

Process finished with exit code 0
```

Рисунок 7. Результат выполнения задания 2

8.Решила индивидуальное задание.

Сумма аргументов, расположенных между первым и вторым положительными аргументами.

```
if __name__ == "__main__":
    print(summa())
    print(summa(1, 2, -3, 3, 4, 1, -5, 3,))
```

ind x

C:\nana\2.10\Scripts\python.exe C:\Users\Elizav

None

8

Рисунок 8. Результат работы индивидуального задания

9. Даны баллы за экзамен студентов в виде ключ-значения, определить средний балл и самый высокий балл среди всех студентов.

```
if __name__ == "__main__":
    marks(
        Миша=60,
        Саша=70,
        Оля=86,
        Коля=30,
        Вова=87,
    )
```

nd2 x

C:\nana\2.10\Scripts\python.exe C:\Users\Elizav

Лучший балл в группе: 87

Средний балл среди всех студентов - 66.6

Рисунок 8. Результат работы индивидуальной задачи

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