

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 8

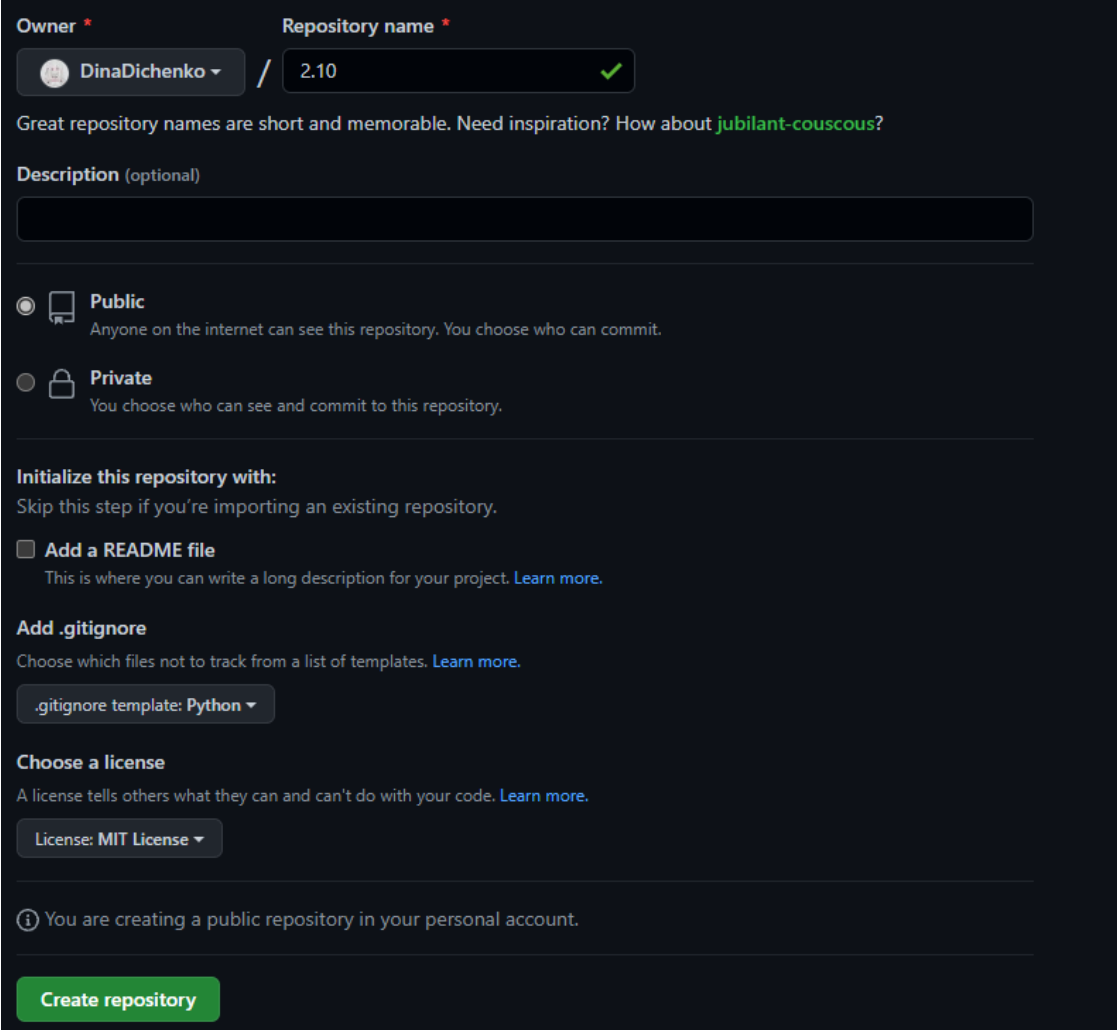
Выполнила: студентка 2
курса, группы ИВТ-б-о-21-1
Диченко Дина Алексеевна

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.



Owner * Repository name *

DinaDichenko / 2_10 ✓

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

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 ▼

i You are creating a public repository in your personal account.

Create repository

Рисунок 1. Создание репозитория

2. Выполнила клонирование созданного репозитория.

```
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python>git clone https://github.com/DinaDichenko/2_10.git
Cloning into '2_10'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python>
```

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 соответствие git-flow.

```
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\2_10>git flow init
Which branch should be used for bringing forth production releases?
- develop
- main
Branch name for production releases: [main] main
Which branch should be used for integration of the "next release"?
- develop
Branch name for "next release" development: [develop] develop
How to name your supporting branch prefixes?
Feature branches? [feature/] fea
Bugfix branches? [bugfix/] bug
Release branches? [release/] rea
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] ver
Hooks and filters directory? [C:/Users/super/Desktop/Dina/ВУЗ/Программирование на python/2_10/.git/hooks] hook
```

Рисунок 4. Организация репозитория в соответствии с git-flow

6. Проработала примеры лабораторной работы.

```
None
6.0
4.5

Process finished with exit code 0
```

Рисунок 5. Результат выполнения примера

7. 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.

```
5. (03013 (Student 07 025 (Program Project  
None  
5.210342169394705  
Process finished with exit code 0
```

Рисунок 6. Результат выполнения задания 1

8. 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.

```
None  
2.18978102189781  
  
Process finished with exit code 0
```

Рисунок 7. Результат выполнения задания 2

9. Решила индивидуальное задание.

Сумму аргументов, расположенных между первым и вторым положительными аргументами.

```
None  
-16  
  
Process finished with exit code 0
```

Рисунок 8. Результат работы индивидуального задания

10. Самостоятельно подобрала или придумала задачу с переменным числом именованных аргументов.

```
Доля черных шаров в бассейне: 16.67 %  
Количество прозрачных шаров: 45  
  
Process finished with exit code 0
```

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.