

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

ОТЧЕТ  
по лабораторной работе №2\_10  
дисциплины  
«Основы программной инженерии»

Выполнил:

Разворотников Денис Сергеевич

2 курс, группа ПИЖ-б-о-21-1,

09.03.04 «Программная инженерия»,

направленность (профиль)

«Разработка и сопровождение  
программного обеспечения», очная  
форма обучения

---

(подпись)

Проверил:

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

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.

1. Был создан репозиторий в Github в который были добавлены правила gitignore для работы IDE PyCharm, была выбрана лицензия MIT, сам репозиторий был клонирован на локальный сервер и был организован в соответствии с моделью ветвления git-flow.

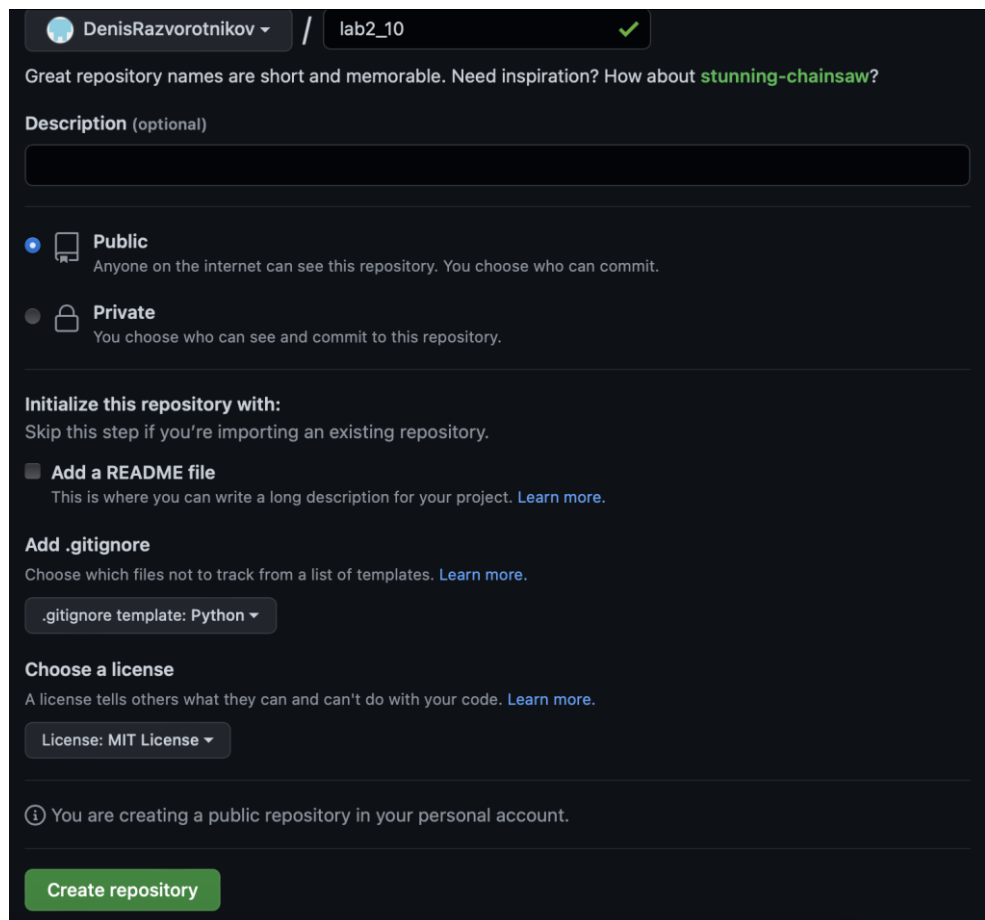
The image shows the GitHub 'Create new repository' page. At the top, the user 'DenisRazvorotnikov' is logged in, and the repository name 'lab2\_10' is entered with a green checkmark. Below this, a message suggests repository names should be short and memorable, with 'stunning-chainsaw?' as an example. A 'Description (optional)' text box is present. The 'Visibility' section has 'Public' selected, with a note that anyone on the internet can see the repository. The 'Initialize this repository with:' section has 'Add a README file' selected. The 'Add .gitignore' section has '.gitignore template: Python' selected. The 'Choose a license' section has 'License: MIT License' selected. At the bottom, a green 'Create repository' button is visible. A note at the bottom states: 'You are creating a public repository in your personal account.'

Рисунок 1 – Создание репозитория

```
denisrazvorotnikov@Air-Denis desktop % git clone https://github.com/DenisRazvorotnikov/lab2_10
Cloning into 'lab2_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.
```

Рисунок 2 – Клонирования репозитория

```
/usr/local/bin/python3 /Users/denisrazvorotnikov/Desktop/lab2_10/1.py
None
6.0
4.5
|
Process finished with exit code 0
```

Рисунок 3 – Результат работы примера

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

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

Код программы:

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

import math

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

if __name__ == "__main__":
    print(f'geometric mean is: {aver_geom(1, 2, 3, 4, 5, 6, 7)}')
    print(f'geometric mean is: {aver_geom()}')
    print(f'geometric mean is: {aver_geom(2.3, 6.5, 9.2, 3.4)}')
```

```
/usr/local/bin/python3 /Users/denisrazvorotnikov/Desktop/lab2_10/2.py
geometric mean is: 3.3800151591412964
geometric mean is: None
geometric mean is: 4.650257311517225

Process finished with exit code 0
```

Рисунок 4 – Результат работы программы

3. Решить поставленную задачу: написать функцию, вычисляющую

среднее гармоническое своих аргументов  $a_1, a_2, \dots, a_n$

Код программы:

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

def aver_harm(*args):
    if args:
        values = [float(arg) for arg in args]
        sz = len(args)
        s = 0
        for arg in values:
            s += 1 / arg
        g = sz / s
        return g
    else:
        return None

if __name__ == "__main__":
    print(f'harmonic mean is: {aver_harm(1, 2, 3, 4, 5, 6, 7, 8)}')
    print(f'harmonic mean is: {aver_harm()}')
    print(f'harmonic mean is: {aver_harm(1.5, 4.6, 9.3, 8.0, 10.5)}')
```

```
harmonic mean is: 2.9434954007884366
harmonic mean is: None
harmonic mean is: 4.126015280007165

Process finished with exit code 0
```

Рисунок 5 – Результат работы программы

4. Было выполнено индивидуальное задание

Вариант 5

Сумму аргументов, расположенных до последнего положительного аргумента.

Код программы:

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

import math

def sum_pos(*args):
    """
    The sum of the arguments located up to the last positive argument
    """
    if args:
        values = [float(arg) for arg in args]
        sum_ = 0
        for num in values:
            if num >= 0:
                sum_ += num
            else:
                break
        return sum_

if __name__ == "__main__":
    print(f'sum positive arguments: {sum_pos(1, 2, 3, 4, 5, -6, 7, 8)}')
    print(f'sum positive arguments: {sum_pos(1.5, 4.6, -9.3, 8.0, 10.5)}')
```

```
/usr/local/bin/python3 /Users/denisrazvorotnikov/Desktop/lab2_10/idz1.py
sum positive arguments: 15.0
sum positive arguments: 6.1

Process finished with exit code 0
```

Рисунок 6 – Результат работы программы

```
[denisrazvorotnikov@Air-Denis lab2_10 % git add .
[denisrazvorotnikov@Air-Denis lab2_10 % git commit -m "Tasks"
[develop 73a68dd] Tasks
10 files changed, 127 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/lab2_10.iml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/vcs.xml
create mode 100644 1.py
create mode 100644 2.py
create mode 100644 3.py
create mode 100644 idz1.py
denisrazvorotnikov@Air-Denis lab2_10 %
```

```

Updating 5517054..73a68dd
Fast-forward
 .idea/.gitignore | 3 +++
 .idea/inspectionProfiles/profiles_settings.xml | 6 ++++++
 .idea/lab2_10.iml | 8 ++++++++
 .idea/misc.xml | 7 ++++++
 .idea/modules.xml | 8 ++++++
 .idea/vcs.xml | 6 ++++++
 1.py | 21 +++++++++++++++++++++++++++++++++++++
 2.py | 23 +++++++++++++++++++++++++++++++++++++
 3.py | 21 +++++++++++++++++++++++++++++++++++++
 idz1.py | 24 +++++++++++++++++++++++++++++++++++++
10 files changed, 127 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/lab2_10.iml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/vcs.xml
create mode 100644 1.py
create mode 100644 2.py
create mode 100644 3.py
create mode 100644 idz1.py
denisrazvorotnikov@Air-Denis lab2_10 %

```

Рисунок 7 – Коммит и пуш изменений переход на ветку main и слияние ее с веткой develop

Вывод: в ходе лабораторной работы были приобретены навыки по работе с функциями с переменным числом параметров при написании программ спомощью языка программирования Python версии 3.x.

### Контрольные вопросы

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 неименованных аргументов функции.

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