

**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: «»

Выполнил: студент 2 курса

группы ИВТ-б-о-21-1

Урусов Максим Андреевич

Ставрополь 2022


Выполнение работы:

1. Создал репозиторий в GitHub «rep 2.6» в который добавил .gitignore, который дополнил правилами для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на лок. сервер и организовал в соответствии с моделью ветвления git-flow.

Create a new repository


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


Owner * **Repository name ***

 MaksimUrusov ▾ / Laba-2.10 ✓

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

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 ▾

This will set  **main** as the default branch. Change the default name in your [settings](#).


 You are creating a public repository in your personal account.

Рисунок 1.1 Создание репозитория

```
C:\Users\den-n>git clone https://github.com/MaksimUrusov/Laba-2.10.git
Cloning into 'Laba-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 1.2 Клонирование репозитория

```
C:\Users\den-n>cd Laba-2.10
C:\Users\den-n\Laba-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/den-n/Laba-2.10/.git/hooks]
```

Рисунок 1.3 Организация репозитория в соответствии с моделью ветвления
git-flow



```
.gitignore – Блокнот
Файл Правка Формат Вид Справка
.idea/
# 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,
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

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

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

Рисунок 1.4 Изменение .gitignore

2. Создал проект PyCharm в папке репозитория, проработал примеры ЛР.

```

None
6.0
4.5

Process finished with exit code 0

```

Рисунок 2.1 Рез-т выполнения программы

3. Выполнил задания.

Задание 1. Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов a_1, a_2, \dots, a_n

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

```

"C:\Users\adamkh\Desktop\3 семестр\Пр
2.213363839400643

Process finished with exit code 0

```

Рисунок 3.1 Вывод программы задания

Задание 2. Решить поставленную задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов

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

```

2.213363839400643
1.9200000000000004

Process finished with exit code 0

```

Рисунок 3.2 Вывод программы задания

4. Выполнил индивидуальное задание.

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 18). Напишите функцию, находящую сумму положительных аргументов, расположенных до максимального аргумента, принимающую произвольное количество аргументов, и возвращающую требуемое значение. Если функции передается пустой список аргументов, то она должна возвращать значение None

Рисунок 4.1 Вывод программы индивидуального задания

5. Сделал коммит, выполнил слияние с веткой main, и запустил изменения в уд. репозиторий.

```
C:\Users\den-n\Laba-2.10>git add .
C:\Users\den-n\Laba-2.10>git commit -m "test1"
[main 56a40a5] test1
4 files changed, 258 insertions(+), 3 deletions(-)
create mode 100644 ind.py
create mode 100644 prim.py
create mode 100644 zadanie.py
C:\Users\den-n\Laba-2.10>git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 4.12 KiB | 1.37 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MaksimUrusov/Laba-2.10.git
b666f3d..56a40a5 main -> main
```

Рисунок 4.1 коммит и пуш изменений и переход на ветку main

```
C:\Users\den-n\Laba-2.10>git checkout main
Switched to branch 'main'
M       .gitignore
A       ind.py
A       prim.py
A       zadanie.py
Your branch is up to date with 'origin/main'.
C:\Users\den-n\Laba-2.10>git merge develop
Already up to date.
```

```

C:\Users\adamkh\Desktop\3 семестр\Программирование на Python\2.10\2.10>git merge develop
Updating e917b11..1125a56
Fast-forward
 .gitignore | 157 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 ind.py      | 26  ++++++
 prim.py     | 23  ++++++
 zadaniya.py | 37  ++++++
 4 files changed, 240 insertions(+), 3 deletions(-)
 create mode 100644 ind.py
 create mode 100644 prim.py
 create mode 100644 zadaniya.py

```

Рисунок 4.2 Слияние ветки main с develop

```

C:\Users\adamkh\Desktop\3 семестр\Программирование на Python\2.10\2.10>git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AdamKh/2.10.git
 e917b11..1125a56  main -> main

```

Рисунок 4.3 Пуш изменений на удаленный сервер

Контр. вопросы и ответы на них:

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 `*`?

Функция также может принимать переменное количество позиционных аргументов, тогда перед именем ставится `*`.

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

Каждая из этих конструкций используется для распаковки аргументов соответствующего типа, позволяя вызывать функции со списком аргументов переменной длины.