

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №2.10 по дисциплине основы программной
инженерии**

Выполнил:
Шальнев Владимир Сергеевич,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:
Доцент кафедры
прикладной математики и
компьютерной безопасности,
Воронкин Р.А.

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2021 г.

ВЫПОЛНЕНИЕ:

```
F:\>cd F:\pythonProject

F:\pythonProject>git clone https://github.com/HAXF13D/laboratory-13
Cloning into 'laboratory-13'...
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
Unpacking objects: 100% (11/11), done.

F:\pythonProject>cd laboratory-13
```

Клонирование репозитория

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4
5      def median(*args):
6          if args:
7              values = [float(arg) for arg in args]
8              values.sort()
9              n = len(values)
10             idx = n // 2
11             if n % 2:
12                 return values[idx]
13             else:
14                 return (values[idx - 1] + values[idx]) / 2
15         else:
16             return None
17
18
19  ▶  if __name__ == "__main__":
20      print(median())
21      print(median(3, 7, 1, 6, 9))
22      print(median(1, 5, 8, 4, 3, 9))
23
```

Пример 1

```
None
6.0
4.5
```

Значение №1

```

1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def geometric_mean(*args):
6     if args:
7         values = [float(arg) for arg in args]
8         n = len(values)
9         result = 1
10        for value in values:
11            result *= value
12        return result ** (1 / n)
13    else:
14        return None
15
16
17 ▶ if __name__ == "__main__":
18     print(geometric_mean())
19     print(geometric_mean(3, 7, 1, 6, 9))
20     print(geometric_mean(1, 5, 8, 4, 3, 9))
21

```

Задача 1

```

None
4.082466323576217
4.0356543087298045

```

Значение №1

```

1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def harmonic_mean(*args):
6     if args:
7         values = [float(arg) for arg in args]
8         n = len(values)
9         result = 0
10        for value in values:
11            result += (1 / value)
12        return n / result
13    else:
14        return None
15
16
17 ▶ if __name__ == "__main__":
18     print(harmonic_mean())
19     print(harmonic_mean(3, 7, 1, 6, 9))
20     print(harmonic_mean(1, 5, 8, 4, 3, 9))
21

```

Задача 2

```
None
2.850678733031674
2.97111416781293
```

Значение №1

```
F:\pythonProject\laboratory-13>git status
On branch develop
Your branch is ahead of 'origin/develop' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   tasks/task_1.py
        new file:   tasks/task_2.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   tasks/task_1.py
        modified:   tasks/task_2.py

F:\pythonProject\laboratory-13>git add .

F:\pythonProject\laboratory-13>git commit -m "Add two tasks"
[develop 8555981] Add two tasks
 2 files changed, 40 insertions(+)
 create mode 100644 tasks/task_1.py
 create mode 100644 tasks/task_2.py
```

Коммит изменений

```
1  ▶  #!/usr/bin/env python3
2      #- coding: utf-8 -*-
3
4
5      def individual_func(*args):
6          if args:
7              result = 0
8              flag = 0
9              for arg in args:
10                 if isinstance(arg, (float, int, str)):
11                     if int(arg) < 0:
12                         flag += 1
13                     elif flag == 1:
14                         result += int(arg)
15                 if flag == 1:
16                     result = 0
17             return result
18         else:
19             return None
20
21
22  ▶  if __name__ == "__main__":
23         print(individual_func())
24         print(individual_func(-3, 7, 1, 6, -9))
25         print(individual_func(1, -5, 8, -4, 3, 9, -4, 5, -6))
26
```

Индивидуальное задание 1

None
14
8

Значение №1

```
F:\pythonProject\laboratory-13>git status
On branch develop
Your branch is ahead of 'origin/develop' by 2 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   tasks/indiv.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   tasks/indiv.py

F:\pythonProject\laboratory-13>git add .
F:\pythonProject\laboratory-13>git commit -m "Add indiv task"
[develop e03f3b2] Add indiv task
1 file changed, 25 insertions(+)
create mode 100644 tasks/indiv.py
```

Коммит изменений

ОТВЕТЫ НА ВОПРОСЫ:

1. Какие аргументы называются позиционными в Python?

Позиции аргументы, это такие аргументы, значение которых будет зависит от их позиции. Пример: `def test(a, b) -> a, b` – это позиционные аргументы.

именно по позиции, расположению аргумента, функция понимает, какому параметру он соответствует.

2. Какие аргументы называются именованными в Python?

Аргументы, передаваемые с именами, называются именованными. При вызове функции можно использовать имена параметров из ее определения.

3. Для чего используется оператор * ?

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

Вот пример:

```
a = [1, 2, 3]
```

```
b = [*a, 4, 5, 6]
```

```
print(b) # [1, 2, 3, 4, 5, 6]
```

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

Итак, мы знаем о том, что оператор «звездочка» 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.