

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 №13 по дисциплине «Основы
программной инженерии»

Выполнил:
Чернова Софья Андреевна,
2 курс, группа ПИЖ-б-о-20-1,
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2021 г

1. Ход работы:
 - 1.1 Пример 1 (рис. 1, 2)

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

Рисунок 1 – код программы

```
None
6.0
4.5

Process finished with exit code 0
```

Рисунок 2 – результат работы программы

- 1.2 Задача 1 (рис. 3, 4, 5)

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      def geom(*args):
5          if args:
6              multi = 1
7              values = [float(arg) for arg in args]
8              n = len(values)
9              for elem in values:
10                 multi *= elem
11             return multi ** (1 / n)
12         else:
13             return None
14
15
16  ▶  if __name__ == '__main__':
17       arguments = [float(i) for i in input("Enter the arguments: ").split()]
18       print(f"The geometric mean of these arguments is: {geom(*arguments)}")

```

Рисунок 3 – код программы

```

Enter the arguments: 5 4 6 4 6 3 7 8
The geometric mean of these arguments is: 5.135558978695357

Process finished with exit code 0

```

Рисунок 4 – результат работы программы

```

Enter the arguments:
The geometric mean of these arguments is: None

Process finished with exit code 0

```

Рисунок 5 – результат работы программы при вводе пустого множества

1.3 Задача 2 (рис. 6, 7, 8)

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4
5  def mid_harm(*args):
6      if args:
7          values = [float(arg) for arg in args]
8          n = len(values)
9          sum_of_reversed = 0
10         for value in values:
11             sum_of_reversed += (1 / value)
12         return n / sum_of_reversed
13     else:
14         return None
15
16
17  ▶  if __name__ == "__main__":
18      arguments = [float(i) for i in input("Enter the arguments: ").split()]
19      print(f"The harmonic mean of these arguments is: {mid_harm(*arguments)}")

```

Рисунок 6 – код программы

```

Enter the arguments: 7 8 5 6 14 38 1
The harmonic mean of these arguments is: 4.040944767967591

Process finished with exit code 0

```

Рисунок 7 – результат работы программы

```

Enter the arguments:
The harmonic mean of these arguments is: None

Process finished with exit code 0

```

Рисунок 8 – результат работы программы при вводе пустого множества

1.4 Задача 3 (рис. 9, 10)

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4
5      def book(author, **books):
6          print(f"Author: {author}")
7          for books, name in books.items():
8              print(f"{name}")
9
10
11  ▶  if __name__ == '__main__':
12      book(
13          "Э. М. Ремарк",
14          book1="На западном фронте без перемен",
15          book2="Триумфальная арка",
16          book3="Три товарища"
17      )
18      book(
19          "М. Ю. Лермонтов",
20          book1="Герой нашего времени"
21      )

```

Рисунок 9 – код программы

```

Author: Э. М. Ремарк
На западном фронте без перемен
Триумфальная арка
Три товарища
Author: М. Ю. Лермонтов
Герой нашего времени

Process finished with exit code 0

```

Рисунок 10 – результат работы программы

1.5 Индивидуальное задание № 6 (рис. 11, 12, 13)

```

1  ▶  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def after_sum(*args):
5      if args:
6          left = 0
7          r = 0
8          mult = 1
9          flagl = False
10         flagr = False
11         for index, arg in enumerate(args):
12             if arg == 0:
13                 if not flagl:
14                     left = index
15                     flagl = True
16                 elif not flagr:
17                     r = index
18                     flagr = True
19             if not (flagr * flagl):
20                 return None
21         else:
22             for index, arg in enumerate(args):
23                 if (index > left) and (index < r):
24                     mult = mult * arg
25             return mult
26         else:
27             return None
28
29
30  ▶  if __name__ == "__main__":
31         arguments = [int(i) for i in input().split()]
32         arguments.reverse()
33         print(after_sum(*arguments))

```

Рисунок 11 – код программы

```

7 0 3 4 5 0 7
60

Process finished with exit code 0

```

Рисунок 12 – результат работы программы

```

5 6 0 4 3 2
None

Process finished with exit code 0

```

Рисунок 13 – результат работы программы при вводе пустого множества

2. Ответы на контрольные вопросы:

1. Какие аргументы называются позиционными в Python?

Позиционные аргументы - это такие аргументы, значение которых будет зависеть 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` ?

Оператор «звёздочка» в Python способен «вытаскивать» из объектов составляющие их элементы. Существует два вида параметров функций, а именно: `*args` — это сокращение от «arguments» (аргументы), а `**kwargs` — сокращение от «keyword arguments» (именованные аргументы).