

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 №13
дисциплины «Программирование на Python»

Выполнил:
Данилецкий Дмитрий Витальевич
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., канд. технических
наук, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

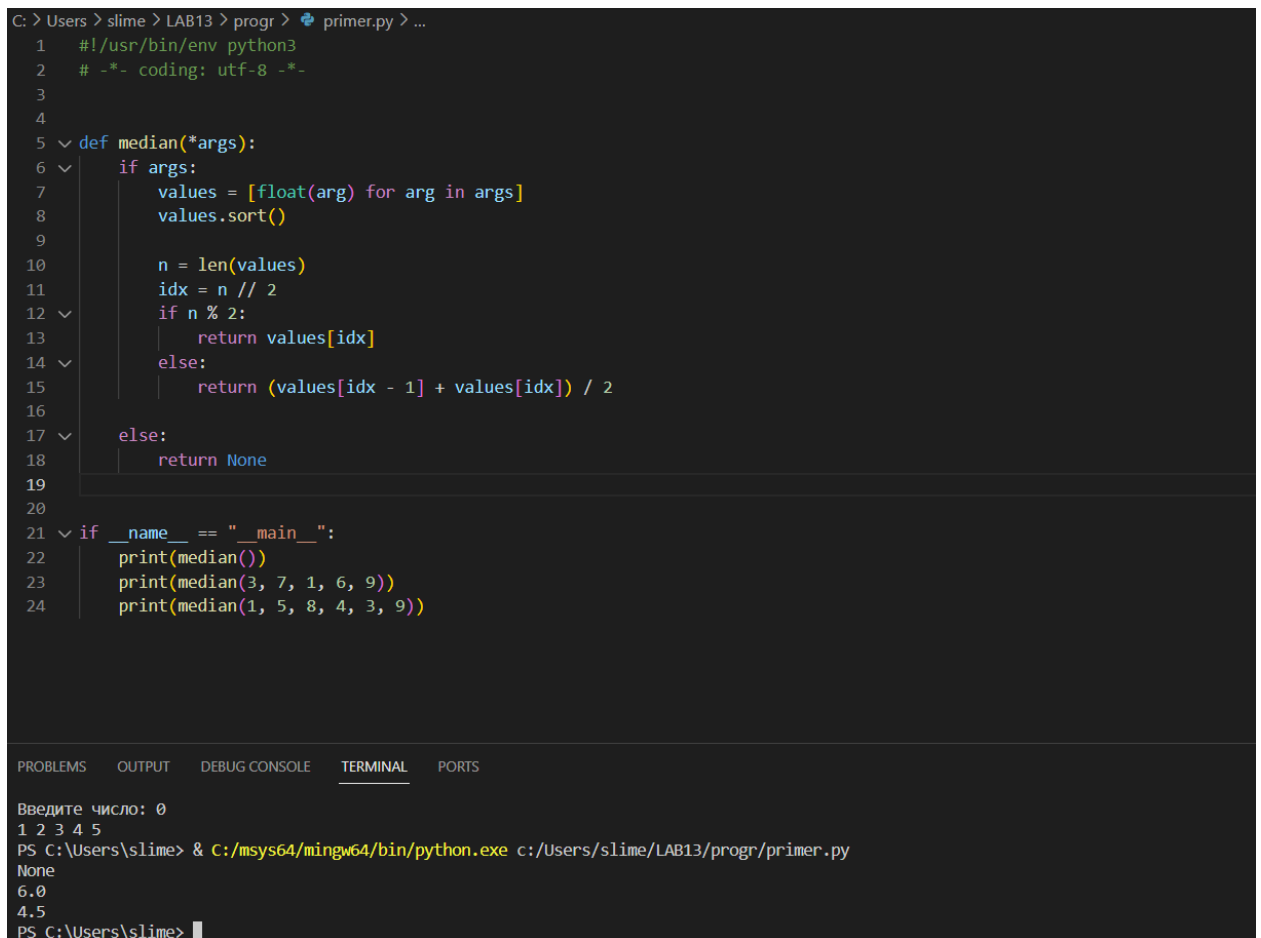
Ставрополь, 2023 г.

Тема: Функции с переменным числом параметров в Python

Цель работы: приобретение навыков по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.

Ход работы

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.
2. Дополнил файл .gitignore необходимыми правилами.
3. Организовал созданный репозиторий в соответствие с моделью ветвления git-flow.
4. Проработал пример лабораторной работы. Создал для него отдельный модуль языка Python. Привел в отчете скриншоты результата выполнения программы примера.



```
C: > Users > slime > LAB13 > progr > primer.py > ...
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
10         n = len(values)
11         idx = n // 2
12         if n % 2:
13             return values[idx]
14         else:
15             return (values[idx - 1] + values[idx]) / 2
16
17     else:
18         return None
19
20
21 if __name__ == "__main__":
22     print(median())
23     print(median(3, 7, 1, 6, 9))
24     print(median(1, 5, 8, 4, 3, 9))

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

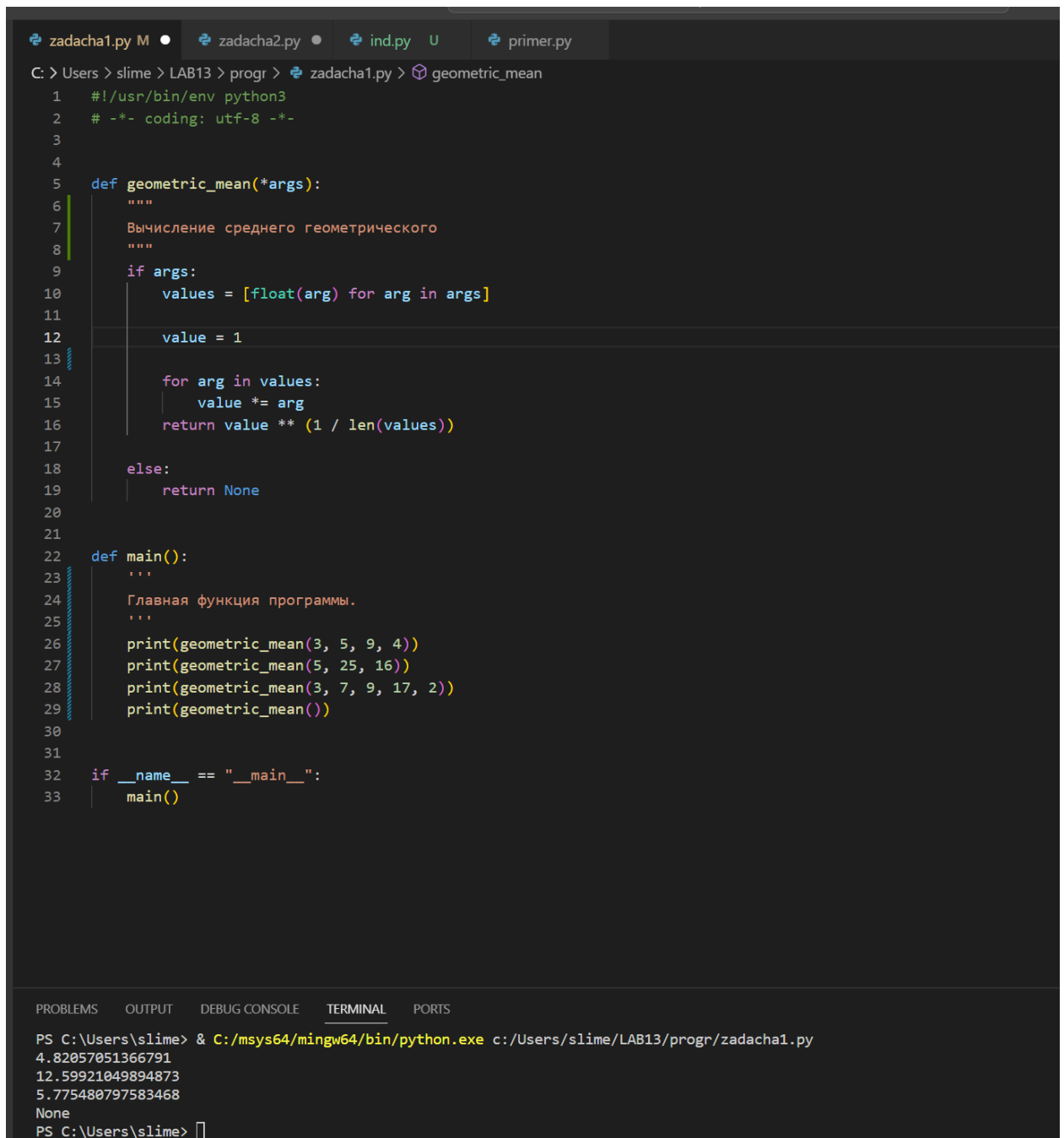
Введите число: 0
1 2 3 4 5
PS C:\Users\slime> & c:\msys64\mingw64\bin\python.exe c:/Users/slime/LAB13/progr/primer.py
None
6.0
4.5
PS C:\Users\slime>
```

Рисунок 1. Результат работы программы из примера 1

5. Решил следующую задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов a_1, a_2, \dots, a_n

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

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



```
zadacha1.py M • zadacha2.py • ind.py U primer.py
C: > Users > slime > LAB13 > progr > zadacha1.py > geometric_mean
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def geometric_mean(*args):
6      """
7      Вычисление среднего геометрического
8      """
9      if args:
10         values = [float(arg) for arg in args]
11
12         value = 1
13
14         for arg in values:
15             value *= arg
16         return value ** (1 / len(values))
17
18     else:
19         return None
20
21
22  def main():
23      """
24      Главная функция программы.
25      """
26      print(geometric_mean(3, 5, 9, 4))
27      print(geometric_mean(5, 25, 16))
28      print(geometric_mean(3, 7, 9, 17, 2))
29      print(geometric_mean())
30
31
32  if __name__ == "__main__":
33      main()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB13/progr/zadacha1.py
4.82057051366791
12.59921049894873
5.775480797583468
None
PS C:\Users\slime>
```

Рисунок 2. Результат работы программы из задачи 1

6. Решил следующую задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов a_1, a_2, \dots, a_n

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

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

```
C: > Users > slime > LAB13 > progr > zadacha2.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def garmonic_mean(*args):
6      """
7      Вычисление среднего гармонического
8      """
9      if args:
10         values = [float(arg) for arg in args]
11
12         reciprocal_sum = sum(1 / arg for arg in values)
13         return len(args) / reciprocal_sum
14
15     else:
16         return None
17
18
19 def main():
20     """
21     Главная функция программы.
22     """
23     print(garmonic_mean(2, 4, 8, 16, 32))
24     print(garmonic_mean(4, 5, 6, 1))
25     print(garmonic_mean(5, 6, 7))
26     print(garmonic_mean())
27
28 if __name__ == "__main__":
29     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB13/progr/zadacha2.py
5.161290322580645
2.4742268041237114
5.88785046728972
None
PS C:\Users\slime> █
```

Рисунок 3. Результат работы программы из задачи 2

7. Выполнил индивидуальное задание, согласно варианту 8. 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. Согласно варианту 8, необходимо найти сумму аргументов, расположенных между первым и вторым положительными аргументами.

```
C: > Users > slime > LAB13 > progr > ind.py > sum_between_positive

5 def sum_between_positive(*args):
6     """
7     Расчет суммы аргументов между первым и вторым положительными аргументами
8     """
9     if args:
10        values = [float(arg) for arg in args]
11
12        first_pos = None
13        second_pos = None
14
15        for i, value in enumerate(values):
16            if value > 0:
17                if first_pos is None:
18                    first_pos = i
19                elif second_pos is None:
20                    second_pos = i
21                break
22
23        if first_pos is None or second_pos is None:
24            return None
25
26        return(sum(values[first_pos + 1:second_pos]))
27
28    else:
29        return None
30
31
32 def main():
33     """
34     Главная функция программы.
35     """
36     print(sum_between_positive(3, -2, -4, 5, -7, 1))
37     print(sum_between_positive(-1, -2, -3, -4, -5))
38     print(sum_between_positive(8, 6, -1, -3, -4, 5))
39     print(sum_between_positive())
40
41 if __name__ == "__main__":
42     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB13/progr/ind.py
-6.0
None
0
None
PS C:\Users\slime>
```

Рисунок 4. Результат работы программы из индивидуального задания

8. Самостоятельно подберите или придумайте задачу с переменным числом именованных аргументов. Приведите решение этой задачи.

Задача: Напишите функцию, которая принимает произвольное количество именованных аргументов, представляющих информацию о книге (например, название, автор, год издания и т. д.). Функция должна собирать эту информацию в словарь и выводить его.

```
C: > Users > slime > LAB13 > progr > mine.py > main
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def book_info(**kwargs):
6      """
7      Собирает информацию о книге в словарь
8      """
9      book_details = {}
10
11     for key, value in kwargs.items():
12         book_details[key] = value
13
14     return book_details
15
16
17 def main():
18     """
19     Главная функция программы
20     """
21     book_details = book_info(
22         title="Дюна",
23         author="Фрэнк Герберт",
24         year=1965,
25         genre="Фантастика",
26         pages=704
27     )
28
29     print(book_details)
30
31 if __name__ == "__main__":
32     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB13/progr/mine.py
{'title': 'Дюна', 'author': 'Фрэнк Герберт', 'year': 1965, 'genre': 'Фантастика', 'pages': 704}
PS C:\Users\slime> █
```

Рисунок 5. Результат работы программы из придуманного задания

Контрольные вопросы

1. Какие аргументы называются позиционными в Python?

В Python аргументы называются позиционными, если они передаются функции в том же порядке, в котором они определены в функции.

В функцию также можно передать переменное количество позиционных аргументов. Это делается с помощью оператора `*` перед именем аргумента в определении функции.

2. Какие аргументы называются именованными в Python?

В Python аргументы называются именованными, если они передаются функции с указанием имени аргумента, за которым следует значение аргумента.

В функцию также можно передать переменное количество именованных аргументов. Это делается с помощью оператора `**` перед именем аргумента в определении функции.

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