

**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**

Выполнила:

Емельянова Яна

Александровна, 2 курс,

группа ПИЖ-б-о-20-1,

Проверил:

Доцент кафедры инфокоммуникаций,

Воронкин Р.А.

Ставрополь, 2021 г.

1. Функции с переменным числом параметров в Python

Примеры из методических указаний

Позиционные и именованные аргументы

```
e1.py x
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4   def print_these(a, b, c):
5       print(a, "is stored in a")
6       print(b, "is stored in b")
7       print(c, "is stored in c")
8
9
10 ▶ if __name__ == "__main__":
11     print_these(1, 2, 3)
12
```

```
↓
1 is stored in a
2 is stored in b
3 is stored in c

Process finished with exit code 0
```

```
e2.py x
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4   def print_these(a, b, c=None):
5       print(a, "is stored in a")
6       print(b, "is stored in b")
7       print(c, "is stored in c")
8
9
10 ▶ if __name__ == "__main__":
11     print_these(1, 2)
12
```

```
↓
1 is stored in a
2 is stored in b
None is stored in c
⌵
⌵
⌵
Process finished with exit code 0
🗑️
```

```
e3.py x
1 ▶️ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def print_these(a=None, b=None, c=None):
5     print(a, "is stored in a")
6     print(b, "is stored in b")
7     print(c, "is stored in c")
8
9
10 ▶️ if __name__ == "__main__":
11     print_these(c=3, a=1)
12
```

```
↓
1 is stored in a
None is stored in b
3 is stored in c
⌵
⌵
⌵
Process finished with exit code 0
🗑️
```

Оператор «звёздочка»

```
e4.py x
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     a = [1, 2, 3]
6     b = [*a, 4, 5, 6]
7     print(b)
8
```

```
↓ [1, 2, 3, 4, 5, 6]
⌵ Process finished with exit code 0
⌵
🖨
🗑
```

Как пользоваться *args и **kwargs ?

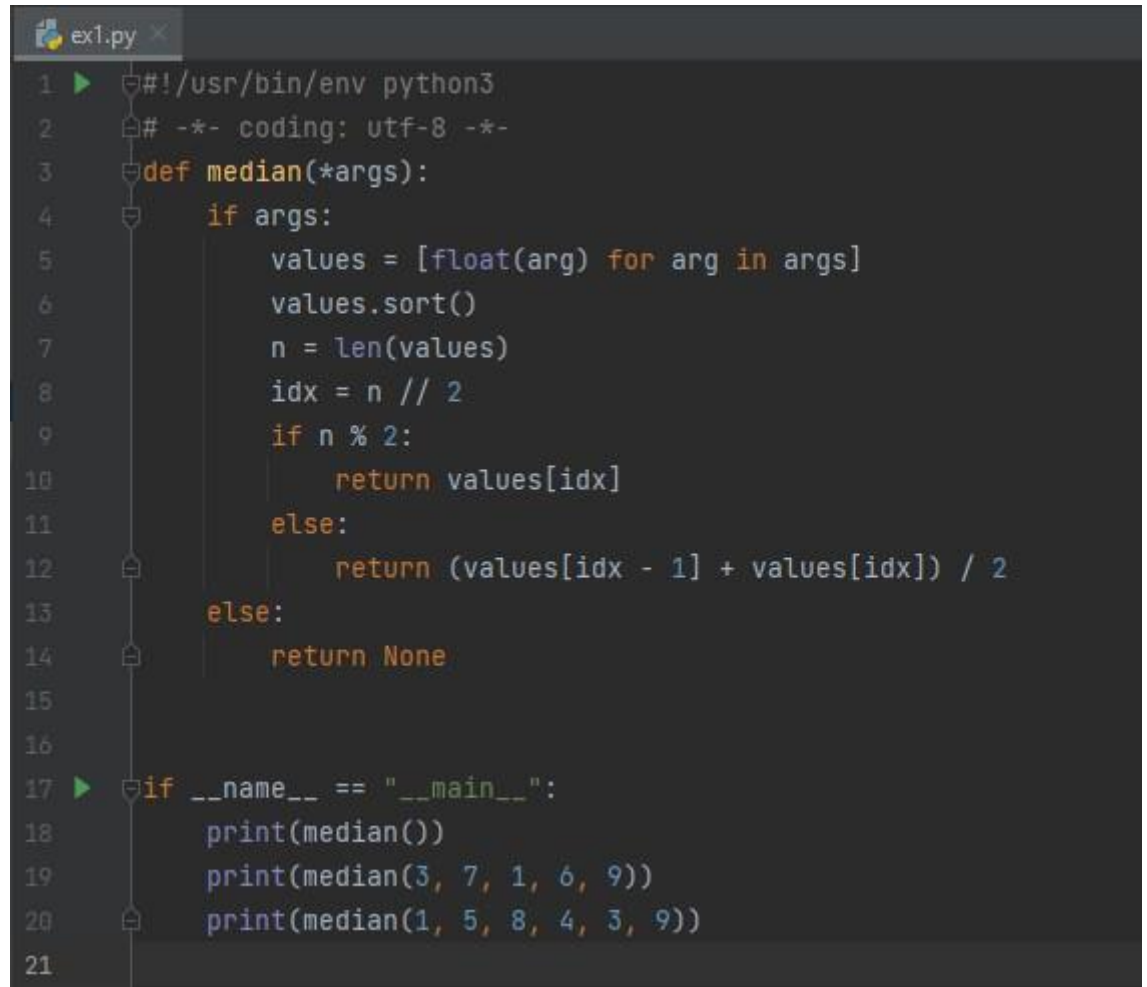
```
e5.py x
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4   def print_scores(student, *scores):
5       print(f"Student Name: {student}")
6       for score in scores:
7           print(score)
8
9
10 ▶ if __name__ == "__main__":
11     ⚡ print_scores("Jonathan", 100, 95, 88, 92, 99)
12
```

```
↓ Student Name: Jonathan
100
95
88
92
99
Process finished with exit code 0
```

```
e6.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def print_pet_names(owner, **pets):
5     print(f"Owner Name: {owner}")
6     for pet, name in pets.items():
7         print(f"{pet}: {name}")
8
9
10 ▶ if __name__ == "__main__":
11     print_pet_names(
12         "Jonathan",
13         dog="Brock", fish=["Larry", "Curly", "Moe"],
14         turtle="Shelldon"
15     )
16
```

```
↓ Owner Name: Jonathan
dog: Brock
fish: ['Larry', 'Curly', 'Moe']
turtle: Shelldon
Process finished with exit code 0
```

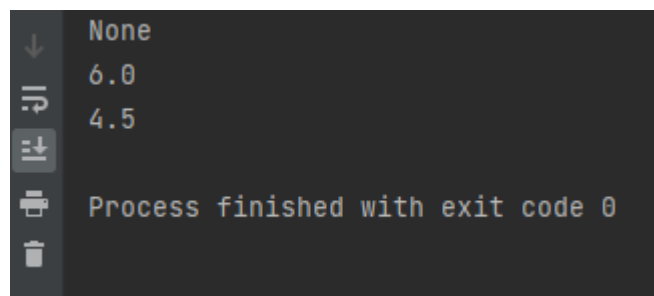
1.1 Пример 1 (рис. 1,2)



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

Рисунок 1 – Код программы

Рисунок 2 – Результат выполнения программы



```
↓ None
↺ 6.0
↺ 4.5
≡
Print Process finished with exit code 0
Trash
```

1.2 Задача 1 (рис. 3-5).

```
z1.py x z2.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def mid_geom(*args):
6     if args:
7         multiplication = 1
8         values = [float(arg) for arg in args]
9         n = len(values)
10        for elem in values:
11            multiplication *= elem
12        return multiplication ** (1 / n)
13    else:
14        return None
15
16
17 ▶ if __name__ == "__main__":
18     arguments = [float(i) for i in input("Введите аргументы: ").split()]
19     print(f"Среднее геометрическое: {mid_geom(*arguments)}")
20
```

Рисунок 3 – Код программы

```
↓ Введите аргументы: 21 4235 21124
⇅ Среднее геометрическое: 1233.908500334683
⇅
⇅ Process finished with exit code 0
```

Рисунок 4 – Результат выполнения программы

```
↓ Введите аргументы:
⇅ Среднее геометрическое: None
⇅
⇅ Process finished with exit code 0
⇅
```

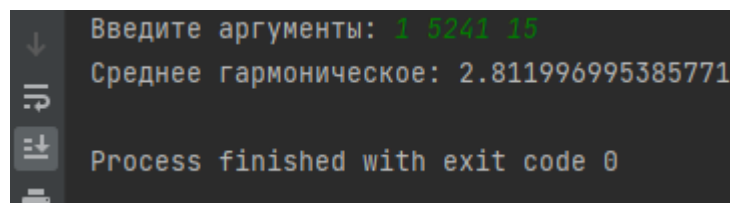
Рисунок 5 – Вывод программы при передаче пустого списка аргументов

1.3 Задача 2 (рис. 6-8).



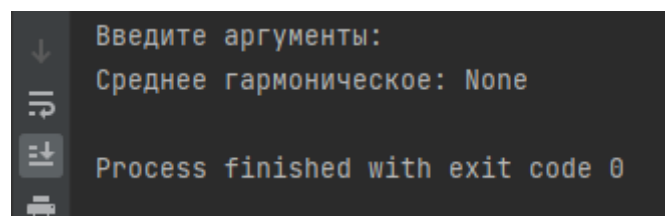
```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def mid_harm(*args):
6      if args:
7          values = [float(arg) for arg in args]
8          n = len(values)
9          sum_of_reversed = 0
10         for value in values:
11             sum_of_reversed += (1 / value)
12         return n / sum_of_reversed
13     else:
14         return None
15
16
17 if __name__ == "__main__":
18     arguments = [float(i) for i in input("Введите аргументы: ").split()]
19     print(f"Среднее гармоническое: {mid_harm(*arguments)}")
20
```

Рисунок 6 – Код программы



```
↓ Введите аргументы: 1 5241 15
⌵ Среднее гармоническое: 2.811996995385771
⌵ Process finished with exit code 0
⌵
```

Рисунок 7 – Вывод программы



```
↓ Введите аргументы:
⌵ Среднее гармоническое: None
⌵ Process finished with exit code 0
⌵
```

Рисунок 8 – Вывод программы при пустых аргументах

Задача 3 (рис. 9-10).

Условие: в функцию подаётся имя судна и вес привезённого груза в тоннах, затем она выводит название судна с максимальным весом груза.


```
z3.py x
1  ▶ #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def min_laboratories(**kwargs):
6      """ Максимальный вес груза, прибывшего в порт.
7
8      На вход поступает название судна и общий вес груза
9
10     Функция выводит название судна, которое прибыло с грузом наибольшего веса
11
12     """
13     if kwargs:
14         max_weight = max(kwargs.values())
15         for vessel, weight in kwargs.items():
16             if weight == max_weight:
17                 print(
18                     f"Наибольший вес груза - ({weight} тонн) "
19                     f"у судна: {vessel}"
20                 )
21     else:
22         return None
23
24
25  ▶ if __name__ == "__main__":
26     min_laboratories(Arkadiy=90, Chayka=4, Napoleon=1, London=7)
27     min_laboratories(Starflight=16, Vansk=5, Alekstour=45)
28     min_laboratories(Flavour=13)
29
```

Рисунок 9 – Код программы

```
↓
Наибольший вес груза - (90 тонн) у судна: Arkadiy
Наибольший вес груза - (45 тонн) у судна: Alekstour
Наибольший вес груза - (13 тонн) у судна: Flavour
Process finished with exit code 0
```

Рисунок 10 – Вывод программы

1.4 Индивидуальное задание (рис. 11-13)

Вариант 9

```
ind_zad.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def zero_sum(*args):
6     if args:
7         i = 0
8         for index, arg in enumerate(args):
9             if arg == 0:
10                 i = index
11                 break
12         zer_s = sum(abs(arg) for index, arg in enumerate(args) if index > i)
13         return zer_s
14     else:
15         return None
16
17
18 ▶ if __name__ == "__main__":
19     arguments = [int(i) for i in input("Введите аргументы: ").split()]
20     print("Сумма модулей аргументов после первого 0: "
21           f"{zero_sum(*arguments)}")
22 )
23
```

Рисунок 11 – Код программы

```
Введите аргументы: 1 426 42742 824 72 0 -14 56
Сумма модулей аргументов после первого 0: 70

Process finished with exit code 0
```

Рисунок 12 – Вывод программы

```
Введите аргументы:
Сумма модулей аргументов после первого 0: None

Process finished with exit code 0
```

Рисунок 13 – Вывод программы при пустой строке аргументов

2. Ответы на контрольные вопросы

1. Какие аргументы называются позиционными в Python?

При вызове функций значения в такие аргументы подставляются согласно позиции имён аргументов в определении функции.

2. Какие аргументы называются именованными в Python?

Это аргументы, передаваемые в функцию вместе с именем.

3. Для чего используется оператор * ?

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

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 ?

*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. Благодаря использованию * мы создаём список позиционных аргументов на основе того, что было передано функции при вызове. Благодаря символам ** создаётся словарь, в котором содержатся именованные аргументы, переданные функции при её вызове.