

Федеральное государственное автономное образовательное учреждение
высшего образования «Северо-Кавказский федеральный университет»
Институт цифрового развития

ОТЧЕТ ПО ДИСЦИПЛИНЕ
ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ
Лабораторная работа №13

Выполнила:

Мирзаева Камилла Мирзаевна

2 курс, группа ПИЖ-б-о-20-1

Принял:

Воронкин Роман Александрович

Ставрополь, 2021 г.

Ход работы:

```
def print_these(a, b, c):  
    print(a, "is stored in a")  
    print(b, "is stored in b")  
    print(c, "is stored in c")
```

Рисунок 1 - функции сопоставления позиции аргументов и параметров

```
> def print_these(a, b, c=None):  
    print(a, "is stored in a")  
    print(b, "is stored in b")  
    print(c, "is stored in c")  
  
    print_these(1, 2)  
  
print_these()
```

main ×

```
1 is stored in a  
2 is stored in b  
None is stored in c
```

Рисунок 2 - Задание опционального параметра

```
> a = [1, 2, 3]  
b = [*a, 4, 5, 6]  
  
print(b)
```

main ×

```
[1, 2, 3, 4, 5, 6]
```

Рисунок 3 - Пример «распаковки» объекта

```
def print_scores(student, *scores):
    print(f"Student Name: {student}")
    for score in scores:
        print(score)

print_scores("Jonathan", 100, 95, 88, 92, 99)
```

main x

Student Name: Jonathan
100
95
88
92
99

Рисунок 4 - Создание функции, которая умеет выводить результаты, набранные учеников в тесте

```
def print_pet_names(owner, **pets):
    print(f"Owner Name: {owner}")
    for pet, name in pets.items():
        print(f"{pet}: {name}")

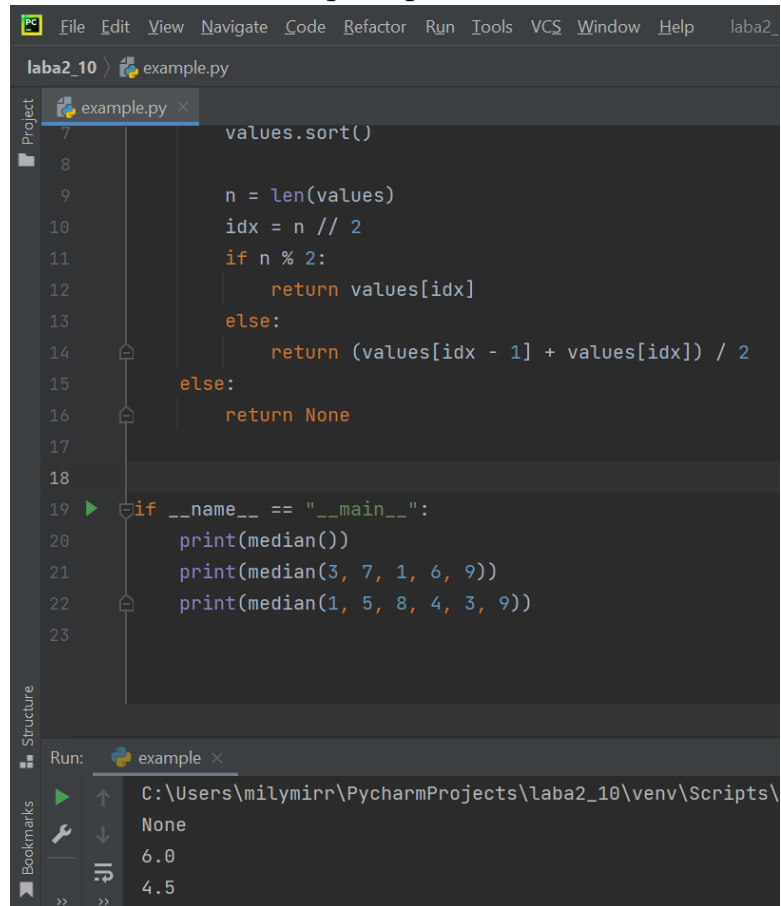
print_pet_names(
    "Jonathan",
    dog="Brock", fish=["Larry", "Curly", "Moe"],
    turtle="Shelldon"
)
```

main x

Owner Name: Jonathan
dog: Brock
fish: ['Larry', 'Curly', 'Moe']
turtle: Shelldon

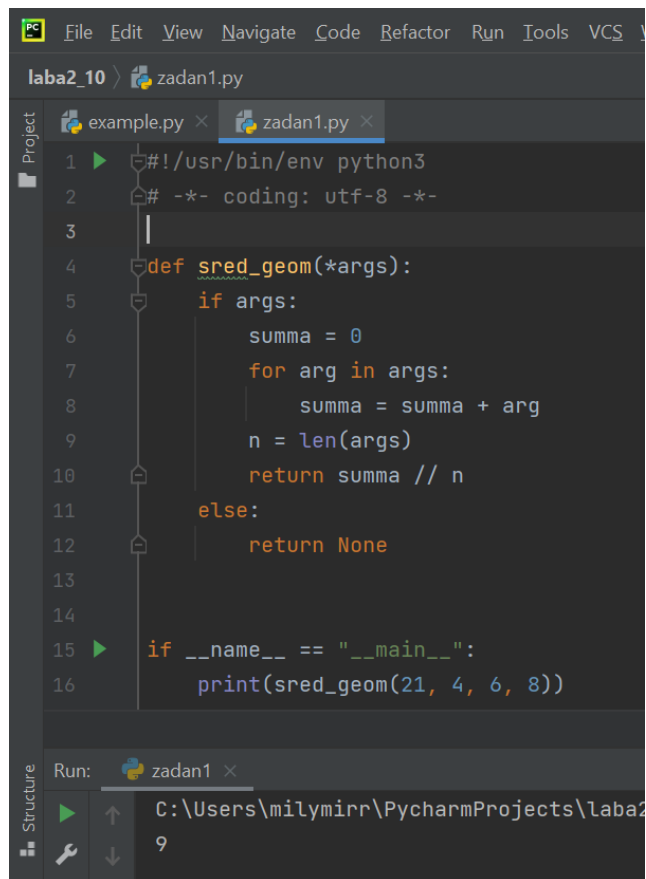
Рисунок 5 - Работа с именованными аргументами

Пример №1



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help lab2_10
lab2_10 > example.py
example.py x
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 if __name__ == "__main__":
20     print(median())
21     print(median(3, 7, 1, 6, 9))
22     print(median(1, 5, 8, 4, 3, 9))
23
Run: example x
C:\Users\milymirr\PycharmProjects\lab2_10\venv\Scripts\
None
6.0
4.5
```

Рисунок 6 - Пример №1



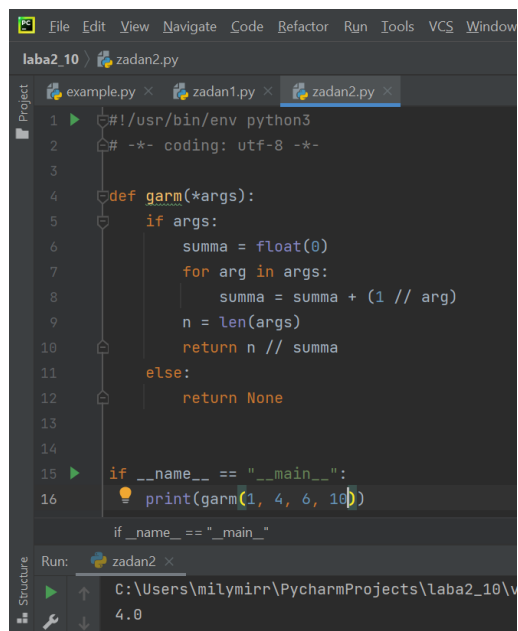
```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def sred_geom(*args):
5     if args:
6         summa = 0
7         for arg in args:
8             summa = summa + arg
9         n = len(args)
10        return summa // n
11    else:
12        return None
13
14
15 if __name__ == "__main__":
16     print(sred_geom(21, 4, 6, 8))
```

Run: zad1 x

C:\Users\milymirr\PycharmProjects\laba2

9

Рисунок 7 - Задание №1



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

Run: zad2 x

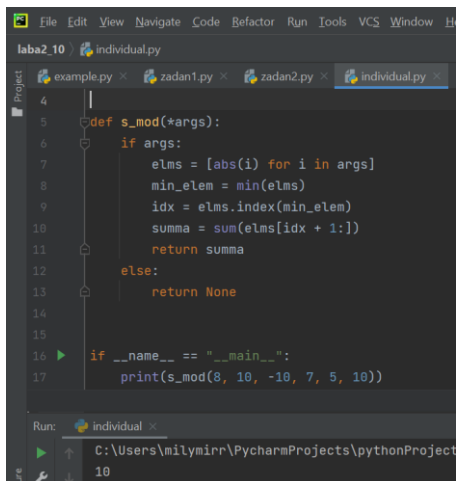
C:\Users\milymirr\PycharmProjects\laba2_10\

4.0

Рисунок 8 - Задание №2

Индивидуальное задание. Вариант 15

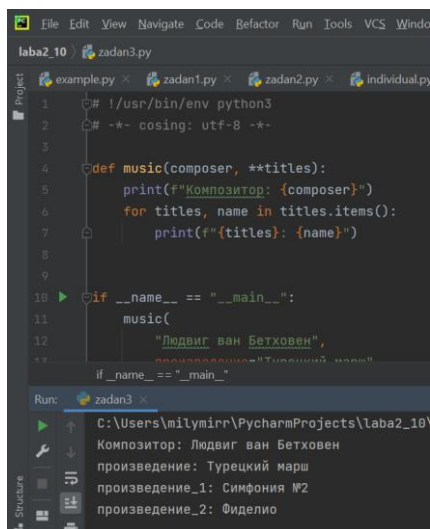
Сумму модулей аргументов, расположенных после минимального по модулю аргумента.



```
4 |
5 | def s_mod(*args):
6 |     if args:
7 |         elms = [abs(i) for i in args]
8 |         min_elem = min(elms)
9 |         idx = elms.index(min_elem)
10 |        summa = sum(elms[idx + 1:])
11 |        return summa
12 |     else:
13 |        return None
14 |
15 |
16 | if __name__ == "__main__":
17 |     print(s_mod(8, 10, -10, 7, 5, 10))
```

Run: individual x
C:\Users\milymirr\PycharmProjects\pythonProject
10

Рисунок 9 - Индивидуальное задание



```
1 | # !/usr/bin/env python3
2 | # -*- coding: utf-8 -*-
3 |
4 | def music(composer, **titles):
5 |     print(f"Композитор: {composer}")
6 |     for titles, name in titles.items():
7 |         print(f"{titles}: {name}")
8 |
9 |
10 | if __name__ == "__main__":
11 |     music(
12 |         "Людвиг ван Бетховен",
13 |         {"Турецкий марш": "Турецкий марш", "Симфония №2": "Симфония №2", "Фиделио": "Фиделио"}
14 |     )
15 | if __name__ == "__main__":
```

Run: zadan3 x
C:\Users\milymirr\PycharmProjects\laba2_10\
Композитор: Людвиг ван Бетховен
произведение: Турецкий марш
произведение_1: Симфония №2
произведение_2: Фиделио

Рисунок 10 - Самостоятельное задание

Контрольные вопросы:

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 *?

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

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

*args используется для передачи произвольного числа именованных аргументов функции.

**kwargs позволяет передавать произвольное число именованных аргументов в функцию.