

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

**Отчёт о лабораторной работе №13 по дисциплине основы программной
инженерии**

Выполнила:

Нестеренко Тамара Антоновна,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:

Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2021 г.

ВЫПОЛНЕНИЕ

1. Практическая часть

```
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 – Пример простейшей 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 x

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

Рисунок 2 – Пример задания опционального параметра

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

main x

[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 ×

Student Name: Jonathan
100
95
88
92
99

Рисунок 4 – Пример создания 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 ×

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

Рисунок 5 – Пример работы с именованными аргументами

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def median(*args):
    if args:
        values = [float(arg) for arg in args]
        values.sort()

        n = len(values)
        idx = n // 2
        if n % 2:
            return values[idx]
        else:
            return (values[idx - 1] + values[idx]) / 2
    else:
        return None

if __name__ == "__main__":
    print(median())
    print(median(3, 7, 1, 6, 9))
    print(median(1, 5, 8, 4, 3, 9))
```

main ×

None
6.0
4.5

Рисунок 6 – Пример решения задачи – разработать функцию для определения медианы значений аргументов функции

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def sred_znach(*args):
    if args:
        summa = 0
        for arg in args:
            summa = summa + arg
        n = len(args)
        return summa // n
    else:
        return None

if __name__ == "__main__":
    print("Среднее значение:", sred_znach(10, 5, 7, 8))
```

main ×

Среднее значение: 7

Рисунок 7 – Пример задания различных значений при работе программы, задание №1

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def sred_znach(*args):
    if args:
        summa = 0
        for arg in args:
            summa = summa + arg
        n = len(args)
        return summa // n
    else:
        return None

if __name__ == "__main__":
    print("Среднее значение:", sred_znach(10, 50, 7, 80, 11, 100))

main x
Среднее значение: 43
```

Рисунок 8 – Пример задания различных значений при работе программы, задание №1

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def sred_znach(*args):
    if args:
        summa = 0
        for arg in args:
            summa = summa + arg
        n = len(args)
        return summa // n
    else:
        return None

if __name__ == "__main__":
    print("Среднее значение:", sred_znach(1, 1, 2, 5, 1, 0))

main x
Среднее значение: 1
```

Рисунок 9 – Пример задания различных значений при работе программы, задание №1

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def sred_znach(*args):
    if args:
        summa = 0
        for arg in args:
            summa = summa + arg
        n = len(args)
        return summa // n
    else:
        return None

if __name__ == "__main__":
    print("Среднее значение:", sred_znach(1, 1, 2, 5, 1, 0))

```

Zadanie_1 ×

Среднее значение: 1

Рисунок 10 – Пример задания различных значений при работе программы, задание №1

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def sred_garm(*args):
    if args:
        summa = float(0)
        for arg in args:
            summa = summa + (1 // arg)
        n = len(args)
        return n // summa
    else:
        return None

if __name__ == "__main__":
    print("Среднее гармоническое значение:", sred_garm(1, 2, 5, 7, 8, 9))

```

main ×

Среднее гармоническое значение: 6.0

Рисунок 11 – Пример задания различных значений при работе программы, задание №2

```
# !/usr/bin/env python3
# -*- coding: utf-8 -*-

def sred_garm(*args):
    if args:
        summa = float(0)
        for arg in args:
            summa = summa + (1 // arg)
        n = len(args)
        return n // summa
    else:
        return None

if __name__ == "__main__":
    print("Среднее гармоническое значение:", sred_garm(1, 5, 1, 5))
```

main ×

Среднее гармоническое значение: 2.0

Рисунок 12 – Пример задания различных значений при работе программы, задание №2

```
# !/usr/bin/env python3
# -*- coding: utf-8 -*-

def sred_garm(*args):
    if args:
        summa = float(0)
        for arg in args:
            summa = summa + (1 // arg)
        n = len(args)
        return n // summa
    else:
        return None

if __name__ == "__main__":
    print("Среднее гармоническое значение:", sred_garm(1, 1, 2, 2, 3, 3))
```

main ×

Среднее гармоническое значение: 3.0

Рисунок 13 – Пример задания различных значений при работе программы, задание №2

```
# !/usr/bin/env python3
# -*- coding: utf-8 -*-

def summ_null(*args):
    if args:
        summa = float(0)
        args = reversed(args)
        for arg in args:
            if arg != 0:
                summa += arg
            else:
                break
        return summa
    else:
        return None

if __name__ == "__main__":
    print("Сумма чисел после последнего нуля:", summ_null(1, 1, 2, 2, 3, 3, 0, 0, 1, 0, 5, 5))
```

main x

Сумма чисел после последнего нуля: 10.0

Рисунок 14 – Пример решения индивидуального задания – вариант №16

2. Самостоятельно придуманное задание

Создайте функцию с использованием именованных переменных, которая будет принимать на ввод значения даты и отдельно информацию о артисте и его произведении, и выводить полученные значения.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

date_info = {'year': input("Введите год:"),
             'month': input("Введите месяц:"),
             'day': input("Введите день:")}
track_info = {'artist': input("Введите ФИО:"),
              'title': input("Введите название произведения:")}
all_info = {**date_info, **track_info}

if __name__ == "__main__":
    print("Выведите информацию о композиторе:", all_info)
```

Рисунок 15 – Пример решения задания, придуманного самостоятельно

3. Вопросы для защиты

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.

Ссылка на репозиторий: https://github.com/tamaranesterenko/Python.LR_13