МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.10

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-20-1
Примаков В. Д. « »_____20__г.
Подпись студента_____
Работа защищена « »_____20__г.
Проверил Воронкин Р.А. _____
(подпись)

ВЫПОЛНЕНИЕ

Пример 1

```python
#!/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))
```

```
C:\Users\vadym\Desktop\laba_2_10\pyPr
None
6.0
4.5

Process finished with exit code 0
```

Задание 1

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


def mid(*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(mid(6, 21, 8, 4))
```

```
9

Process finished with exit code 0
```

Задание 2

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

def 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(garm(1, 4, 6, 10))
```

```
C:\Users\vadym\Desktop\laba_2_10\pyProj
4.0

Process finished with exit code 0
```

Задание 3

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

def music(artist, **titles):
    print(f"Artist: {artist}")
    for titles, name in titles.items():
        print(f"{titles}: {name}")


if __name__ == "__main__":
    music(
        "Prodigy",
        Track_1="Breathe",
        Track_2="Smack My Bitch Up",
        Track_3="Firestarter"
    )
```

```
Artist: Prodigy
Track_1: Breathe
Track_2: Smack My Bitch Up
Track_3: Firestarter

Process finished with exit code 0
```

Индивидуальное задание

18. Сумму положительных аргументов, расположенных до максимального аргумента.

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


def sum(*nums):
    if not nums:
        return None
    else:
        numbers = [int(num) for num in nums]
        max = -1
        ind = int
        sum = 0
        for i, num in enumerate(numbers):
            if num > max:
                ind = i
        for i, num in enumerate(numbers):
            if (i != ind) and (num > 0):
                sum += num
        return sum


if __name__ == "__main__":
    print(sum())
```

```
None

Process finished with exit code 0
```

```
23      print(sum(1, 2, -3, 7, 10))
24
        if __name__ == "__main__"
ind
C:\Users\vadym\Desktop\laba_2_10\pyProj\venv\Scripts\python.exe C:/Users/va
10

Process finished with exit code 0
```

Ссылки на репозитории

GitHub - https://github.com/surai5a/laba_2_10

Ответы на контрольные вопросы

1. Позиционные аргументы - 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. Именованные аргументы - 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 *.
3. Оператор * необходим для распаковки итерируемых элементов, передаваемых в качестве аргумента функции.
4. Конструкции * и ** необходимы для распаковки аргументов соответствующего типа.