

## Лабораторная работа №4 Функции в Питоне

### 1. Цель и порядок работы

Цель работы – изучить возможности работы с функциями в Питоне.

Порядок выполнения работы:

Самостоятельно проработать теоретические основы по языку программирования Питон, выполняя указанные в тексте примеры.

Выполнить индивидуальное задание.

### 2. Краткая теория

#### Функции в Python

Функция - это блок организованного, многократно используемого кода, который используется для выполнения конкретного задания. Функции обеспечивают лучшую модульность приложения и значительно повышают уровень повторного использования кода.

Существуют некоторые правила для создания функций в Python.

Блок функции начинается с ключевого слова `def`, после которого следуют название функции и круглые скобки `()`.

Любые аргументы, которые принимает функция должны находиться внутри этих скобок. После скобок идет двоеточие `:` и с новой строки с отступом начинается тело функции.

*Пример функции в Python:*

```
def my_function(argument):  
    print argument
```

#### Вызов функции

После создания функции, ее можно исполнять, вызывая из другой функции или напрямую из оболочки Python. Для вызова функции следует ввести ее имя и добавить скобки.

Например:

```
my_function("abracadabra")
```

#### Аргументы функции в Python

Вызывая функцию, мы можем передавать ей следующие типы аргументов:

Обязательные аргументы (Required arguments)

Аргументы-ключевые слова (Keyword argument)

Аргументы по умолчанию (Default argument)

Аргументы произвольной длины (Variable-length arguments)

Если при создании функции мы указали количество передаваемых ей аргументов и их порядок, то и вызывать ее мы должны с тем же количеством аргументов, заданных в нужном порядке.

Например:

```
def bigger(a,b):  
    if a > b:  
        print a  
    else:  
        print b  
#Корректное использование функции  
bigger(5,6)  
#Некорректное использование функции  
bigger(3)  
bigger(12,7,3)
```

Аргументы - ключевые слова используются при вызове функции. Благодаря ключевым аргументам, вы можете задавать произвольный (то есть не такой каким он описан при создании функции) порядок аргументов.

Например:

```
def person(name, age):  
    print name, "is", age, "years old"  
#Хотя в описании функции первым аргументом идет имя, мы можем  
#вызвать функцию вот так  
person(age=23, name="John")
```

Аргумент по умолчанию, это аргумент, значение для которого задано изначально, при создании функции.

Например:

```
def space(planet_name, center="Star"):  
    print planet_name, "is orbiting a", center  
#Можно вызвать функцию space так:  
space("Mars")  
#В результате получим: Mars is orbiting a Star  
#Можно вызвать функцию space иначе:  
space("Mars", "Black Hole")  
#В результате получим: Mars is orbiting a Black Hole
```

Иногда возникает ситуация, когда вы заранее не знаете, какое количество аргументов будет необходимо принять функции. В этом случае следует использовать аргументы произвольной длины. Они задаются произвольным именем переменной, перед которой ставится звездочка (\*).

Например:

```
def unknown(*args):  
    for argument in  
        args: print  
            argument  
unknown("hello","world") #напечатает оба слова, каждое с новой строки  
unknown(1,2,3,4,5) # напечатает все числа, каждое с новой строки  
unknown() # ничего не выведет
```

### Ключевое слово return

Выражение return прекращает выполнение функции и возвращает указанное после выражения значение. Выражение return без аргументов это то же самое, что и выражение return None. Соответственно, теперь становится возможным, например, присваивать результат выполнения функции какой либо переменной.

Например:

```
def bigger(a,b):  
    if a > b:  
        return a # Если a больше чем b, то возвращаем b  
                # и прекращаем выполнение функции  
    return b # Незачем использовать else. Если мы дошли  
            # до этой строки, то b, точно не меньше чем a  
# присваиваем результат функции bigger переменной  
num num = bigger(23,42)
```

### Область видимости

Некоторые переменные скрипта могут быть недоступны некоторым областям программы. Все зависит от того, где вы объявили эти переменные.

В Python две базовых области видимости переменных:

- Глобальные переменные.
- Локальные переменные.

Переменные объявленные внутри тела функции имеют локальную область видимости, те что объявлены вне какой-либо функции имеют глобальную область видимости.

Это означает, что доступ к локальным переменным имеют только те функции, в которых они были объявлены, в то время как доступ к глобальным переменным можно получить по всей программе в любой функции.

Например:

```
# глобальная переменная
```

```

age = 44
def info():
    print age # Печатаем глобальную переменную age
def local_info():
    age = 22 # создаем локальную переменную
    print age
info() # напечатает 44
local_info() # напечатает 22

```

Важно помнить, что для того чтобы получить доступ к глобальной переменной, достаточно лишь указать ее имя. Однако, если перед нами стоит задача изменить глобальную переменную внутри функции - необходимо использовать ключевое слово `global`.

Например:

```

#глобальная переменная
age = 13
#функция изменяющая глобальную переменную
def get_older():
    global age
    age += 1
print age # напечатает 13
get_older() # увеличиваем age на 1
print age # напечатает 14

```

### Рекурсия

Рекурсией в программировании называется ситуация, в которой функция вызывает саму себя. Классическим примером рекурсии может послужить функция вычисления факториала числа.

Напомним, что факториалом числа, например, 5 является произведение всех натуральных (целых) чисел от 1 до 5. То есть,  $1 * 2 * 3 * 4 * 5$

Рекурсивная функция вычисления факториала на языке Python будет выглядеть так:

```

def fact(num):
    if num == 0:
        return 1 # Факториал нуля равен единице
    else:
        return num * fact(num - 1)
        # возвращаем результат произведения num
        #и результата возвращенного функцией fact(num - 1)

```

Однако следует помнить, что использование рекурсии часто может быть неоправданным. Дело в том, что в момент вызова функции в оперативной памяти компьютера резервируется определенное количество памяти, соответственно чем больше функций одновременно мы запускаем - тем больше памяти потребуется, что может привести к переполнению стека (stack overflow) и программа завершится аварийно, не так как предполагалось. Учитывая это, там где это возможно, вместо рекурсии лучше применять циклы.

### Анонимные функции, инструкция lambda

Анонимные функции могут содержать лишь одно выражение, но и выполняются они быстрее. Анонимные функции создаются с помощью инструкции lambda. Кроме этого, их не обязательно присваивать переменной, как делали мы инструкцией def func():

```
>>>func = lambda x, y: x + y
>>>func(1, 2)
3
>>>func('a', 'b')
'ab'
>>>(lambda x, y: x + y)(1, 2)
3
>>>(lambda x, y: x + y)('a', 'b')
'ab'
```

lambda функции, в отличие от обычной, не требуется инструкция return, а в остальном, ведет себя точно так же:

```
>>>func = lambda *args: args
>>>func(1, 2, 3, 4)
(1, 2, 3, 4)
```

### **3 Контрольные вопросы**

1. Определение функции?
2. Объявление функции?
3. Какие типы аргументов существуют у функции?
4. Как осуществляется вызов функции?
5. Что такое анонимные функция?
6. Что такое рекурсия?
7. Какие виды рекурсии вы знаете?

## 4 Задание

1. Написать программу в соответствии с вариантом задания из пункта 5.1.
2. Отладить и протестировать программу.
3. Написать программу в соответствии с вариантом задания из пункта 5.2.
4. Отладить и протестировать программу.
5. Написать программу в соответствии с вариантом задания из пункта 5.3.
6. Отладить и протестировать программу.
7. Написать программу в соответствии с вариантом задания из пункта 5.4.
8. Отладить и протестировать программу.

## 5 Варианты заданий

### 5.1 Функции, параметры функций

Варианты заданий выбирать согласно номера в списке группы.

Определить функции, выполняющие действия в соответствии с вариантом задания, по одной на каждый способ передачи параметров. Написать программу, осуществляющую вызов этих функций несколько раз с различными типами параметров.

1. Вычислить с использованием подпрограммы – функции  $Z = \text{НОД}(a,b) + \text{НОК}(a,b)$ , где  $a, b$  – целые положительные числа, НОД – наибольший общий делитель, НОК – наименьшее общее кратное.
2. Определить функцию нахождения расстояния между точками. Во множестве точек на плоскости найти пару точек с максимальным расстоянием между ними.
3. Найти наибольшую из высот треугольника. Известны две стороны треугольника и угол между ними.
4. Найти:  $y = \text{среднее}(a,b,c) / \min(a,b,c)$ .
5. Даны действительные числа  $s, t$ . Получить  $g(1.2, s) + g(t, s) - g(2s - 1, st)$ , где  $g(a, b) = \frac{a^2 - b^2}{2 \cdot a \cdot b - a - b} + (a + b) \cdot \sqrt{\frac{|a + b|}{2}}$

6. Вычислить сумму значений функций

$$Z = f(\sin(x) + \cos(y), x + y, 2) + f(x \cdot y, \sin(x), \cos(y)) + f(\sin^2(x), x - y^2, y - x^2)$$

где

$$f(u, v, t) = \begin{cases} v \cdot u \cdot t, & \text{если } u > 1 \\ u + t + v, & \text{если } 0 \leq u \leq 1 \\ v - t - u, & \text{если } 0 < u \end{cases}$$

7. Вычислить площадь треугольника, если известны его стороны.

8. Даны действительные числа s, t. Получить  $g(1.2, s) + g(t, s) - g(2s -$

$$1, st), \text{ где } g(a, b) = \frac{a^2 + b^2 - 4 \cdot a \cdot b}{a^2 + 5 \cdot a \cdot b + 3 \cdot b^2 + 4 \cdot a - b}$$

9. Составить программу вычисления суммы квадратов простых чисел, лежащих в интервале [M, N].

10. Даны отрезки a, b, c и d. Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь данного треугольника. (Определить функцию, вычисляющую площадь треугольника, если она существует)

11. Определить функцию нахождения расстояния между точками. Во множестве точек на плоскости найти пару точек с минимальным расстоянием между ними.

12. Найти:  $y = \min(a, b, c) / \max(a, b, c)$ .

13. Вычислить сумму значений функций

$$Z = f(\sin(x) + \cos(y), x + y) + f(\sin(x), \cos(y)) + f(\sin^2(x) - 2, a + b^2)$$

Где

$$f(u, t) = \begin{cases} u + t, & \text{если } u > 1 \\ u - t, & \text{если } 0 \leq u \leq 1 \\ t - u, & \text{если } 0 < u \end{cases}$$

14. Даны значения a и b, найти их среднее арифметическое, среднее геометрическое.

15. Найти:  $y = \max(a, b, c) + \min(a, b, c)$ .

16. Вычислить с использованием подпрограммы – функции  $Z = \text{НОК}(a+b, a \cdot b) + \text{НОК}(a, b)$ , где a, b – целые положительные числа, НОД – наибольший общий делитель, НОК – наименьшее общее кратное.

17. Вычислить среднее геометрическое шести вводимых чисел.

18. Вычислить сумму значений функции  $Z = F(a, b) + F(a^2, b^2) + F(a^2 -$

$$F(u, t) = \begin{cases} u^2 + t^2, & \text{если } u > 0, \quad t > 0 \\ u + t^2, & \text{если } u \leq 0, \quad t \leq 0 \\ u - t, & \text{если } u > 0, \quad t \leq 0 \\ u + t, & \text{если } u \leq 0, \quad t > 0 \end{cases}$$

1, b) + F(a - b, b)

19. Даны действительные числа  $s, t$ . Получить  $f(t, -2s, 1.17) +$

$$f(a, b, c) = \frac{2 \cdot a - b - \sin(c) + a \cdot b}{1 + |c + a|}$$

$f(2.2, t, s-t)$ , где

20. Найти:  $y = \max(a, b, c, d) \cdot \min(a, b, c, d)$ .

21. Вычислить с использованием подпрограммы

$$Z = \frac{\sum_{i=1}^{40} \sin(x_i) + \sum_{i=1}^{40} \cos(y_i)}{\sum_{i=1}^{40} |x_i|}.$$

Каждую сумму вычислять с

использованием одной подпрограммы.

22. Задано множество точек на плоскости. Найти сумму длин отрезков между ними.

23. Вычислить с использованием функции

$Z = \text{НОД}(a, b) + \text{НОД}(a \cdot b, a + b)$ , где  $a, b$  – целые положительные числа, НОД – наибольший общий делитель.

24. Вычислить сумму значений функции

$$Z = f(\sqrt{|x|}, y) + f(a, b) + f(\sqrt{|x|} + 1, -y) + f((|x| - |y|), x),$$

$$\text{где } f(u, t) = \begin{cases} u + 2t, & \text{если } u \geq 0 \\ u + t, & \text{если } u \leq -1 \\ u^2 - 2t + 1, & \text{если } -1 < u < 0 \end{cases}$$

25. Вычислить среднее арифметическое четырех вводимых чисел.

## 5.2 Передача массивов в функцию

Варианты заданий выбирать согласно номера в списке группы.

Определить функции, выполняющие действия в соответствии с вариантом задания.

1. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.

1.1. Найти максимальный положительный элемент.

1.2. Вычислить сумму элементов массива.

2. Дан одномерный массив, состоящий из  $N$  вещественных элементов.

2.1. Найти максимальный элемент.

2.2. Вычислить среднеарифметическое отрицательных элементов массива.

3. Дан одномерный массив, состоящий из  $N$  вещественных элементов.

3.1. Найти минимальный элемент.

3.2. Вычислить произведение не нулевых элементов массива.



4. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
  - 4.1. Найти минимальный положительный элемент.
  - 4.2. Вычислить сумму положительных элементов массива, кратных 3.
5. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
  - 5.1. Найти максимальный положительный элемент.
  - 5.2. Вычислить произведение элементов массива.
6. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
  - 6.1. Найти максимальный элемент.
  - 6.2. Вычислить сумму четных элементов массива.
7. Дан одномерный массив, состоящий из  $N$  вещественных элементов.
  - 7.1. Найти минимальный отрицательный элемент.
  - 7.2. Вычислить среднеарифметическое положительных элементов массива.
8. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
  - 8.1. Найти максимальный элемент.
  - 8.2. Вычислить среднеарифметическое элементов массива.
9. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
  - 9.1. Найти минимальный элемент.
  - 9.2. Вычислить сумму элементов массива.
10. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
  - 10.1. Найти максимальный отрицательный элемент.
  - 10.2. Вычислить произведение отрицательных элементов массива.
11. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
  - 11.1. Найти максимальный элемент.
  - 11.2. Вычислить среднеарифметическое нечетных элементов массива.
12. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
  - 12.1. Найти минимальный положительный элемент.
  - 12.2. Вычислить сумму четных элементов массива.
13. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
  - 13.1. Найти минимальный отрицательный элемент.

- 13.2. Вычислить произведение ненулевых элементов массива, кратных 3.
14. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
- 14.1. Найти максимальный отрицательный элемент.
- 14.2. Вычислить среднеарифметическое четных элементов массива.
15. Дан одномерный массив, состоящий из  $N$  вещественных элементов.
- 15.1. Найти максимальный элемент.
- 15.2. Вычислить среднеарифметическое положительных элементов массива.
16. Дан одномерный массив, состоящий из  $N$  вещественных элементов.
- 16.1. Найти минимальный положительный элемент.
- 16.2. Вычислить произведение не нулевых элементов массива.
17. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
- 17.1. Найти максимальный отрицательный элемент.
- 17.2. Вычислить сумму отрицательных элементов массива.
18. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
- 18.1. Найти минимальный элемент.
- 18.2. Вычислить сумму положительных нечетных элементов массива.
19. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
- 19.1. Найти минимальный положительный элемент.
- 19.2. Вычислить произведение нечетных элементов массива.
20. Дан одномерный массив, состоящий из  $N$  вещественных элементов.
- 20.1. Найти максимальный элемент.
- 20.2. Вычислить среднеарифметическое отрицательных элементов массива.
21. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
- 21.1. Найти максимальный положительный элемент.
- 21.2. Вычислить сумму положительных четных элементов массива.
22. Дан одномерный массив, состоящий из  $N$  целочисленных элементов.
- 22.1. Найти минимальный элемент.
- 22.2. Вычислить произведение ненулевых нечетных элементов массива.

23. Дан одномерный массив, состоящий из N вещественных элементов.
  - 23.1. Найти минимальный положительный элемент.
  - 23.2. Вычислить среднеарифметическое отрицательных элементов массива.
24. Дан одномерный массив, состоящий из N целочисленных элементов.
  - 24.1. Найти максимальный отрицательный элемент.
  - 24.2. Вычислить среднеарифметическое нечетных элементов массива.
25. Дан одномерный массив, состоящий из N целочисленных элементов.
  - 25.1. Найти минимальный отрицательный элемент.
  - 25.2. Вычислить сумму нечетных отрицательных элементов массива.

### 5.3 Передача массивов в функцию

Написать программу, выполняющую действия в соответствии с вариантом задания и передающую массив в функцию. Ввод и вывод массивов выполнить в отдельных функциях.

Вариант определяется по последней цифре в списке студентов группы.

1. Вычислить с использованием функции наименьшие элементы в строке и сумму номеров строк и столбцов, в которых они расположены, для матрицы A(10,15). Результаты формировать в одномерных массивах M(10) и S(10).
2. Дан массив a(8,5). С использованием функции найти среднеквадратичное значение положительных элементов каждой строки массива и сформировать из них одномерный массив b(8).
3. Вычислить с использованием функции max элементы каждой строки матрицы A(10,20). Результаты формировать в одномерных массивах C(10) и D(10).
4. Даны массивы a(3,4), b(2,5). Найти  $Z = (Ma + Mb) / (da + db)$ , где Ma, Mb - среднеарифметические значения массивов A, B. da, db - максимальные отклонения от среднеарифметических значений.
5. Дана матрица A(5,5). Сформировать одномерный массив C(5) из среднегеометрических значений положительных элементов каждого столбца матрицы.

$$Z = \frac{x_{\max} - y_{\min}}{x_{\min} - y_{\max}}$$

6. Вычислить  $Z = \frac{x_{\max} - y_{\min}}{x_{\min} - y_{\max}}$  с использованием функции, где  $x_{\max}$ ,  $x_{\min}$ ,  $y_{\max}$ ,  $y_{\min}$  – максимальные и минимальные элементы соответственно массива  $x(5,2)$  и массива  $y(3,4)$ .
7. Дана матрица  $A(4,5)$ ,  $B(5,6)$ . Вычислить  $Z = p_a + p_b$ , где  $p = \sum_{i=1}^N \max\{x_{ij}\}_{j=1, N}$  сумма максимальных элементов каждой строки матрицы.
8. Вычислить с использованием функции  $\min$  элементы каждой строки матрицы  $A(10,20)$ . Результаты формировать в одномерных массивах  $C(10)$  и  $D(10)$ .
9. Преобразовать массив  $x(3,3)$  в  $y$ , оставив в нем только положительные элементы. Вместо остальных элементов записать 0.
10. Определить количество положительных, отрицательных и нулевых элементов матрицы  $A(10,15)$ . (Создать три функции для нахождения этих значений).

## 5.4 Использование рекурсии

Написать программу, рекурсивно вычисляющую сумму.

Вариант определяется по последней цифре в списке студентов группы.

1. Найти сумму ряда с точностью  $\varepsilon$ , общий член которого равен  $a_n = \frac{n!}{5^n}$ . Точность считается достигнутой, если следующий член последовательности меньше заданного  $\varepsilon$ .
2. Найти сумму ряда с точностью  $\varepsilon$ , общий член которого равен  $a_n = \frac{1}{2^n} + \frac{1}{3^n}$ . Точность считается достигнутой, если следующий член последовательности меньше заданного  $\varepsilon$ .
3. Найти сумму ряда с точностью  $\varepsilon$ , общий член которого равен  $a_n = \frac{(-1)^{n-1}}{n^n}$ . Точность считается достигнутой, если следующий член последовательности меньше заданного  $\varepsilon$ .
4. Найти сумму ряда с точностью  $\varepsilon$ , общий член которого равен  $a_n = \frac{1}{(3n-2)(3n+1)}$ . Точность считается достигнутой, если следующий член последовательности меньше заданного  $\varepsilon$ .
5. Найти сумму ряда с точностью  $\varepsilon$ , общий член которого равен  $a_n = n^2 e^{-\sqrt{n}}$ . Точность считается достигнутой, если следующий член последовательности меньше заданного  $\varepsilon$ .

6. Найти сумму ряда с точностью  $\varepsilon$ , общий член которого равен  $a_n = \frac{n!}{(2n)!}$ . Точность считается достигнутой, если следующий член последовательности меньше заданного  $\varepsilon$ .
7. Найти сумму ряда с точностью  $\varepsilon$ , общий член которого равен  $a_n = \frac{3 \cdot n!}{(2n)!}$ . Точность считается достигнутой, если следующий член последовательности меньше заданного  $\varepsilon$ .
8. Найти сумму ряда с точностью  $\varepsilon$ , общий член которого равен  $a_n = \frac{\ln(n!)}{n^2}$ . Точность считается достигнутой, если следующий член последовательности меньше заданного  $\varepsilon$ .
9. Найти сумму ряда с точностью  $\varepsilon$ , общий член которого равен  $a_n = \frac{(2n-1)}{2^n}$ . Точность считается достигнутой, если следующий член последовательности меньше заданного  $\varepsilon$ .
10. Найти сумму ряда с точностью  $\varepsilon$ , общий член которого равен  $a_n = \frac{10^n}{(2n)!}$ . Точность считается достигнутой, если следующий член последовательности меньше заданного  $\varepsilon$ .