

Лабораторная работа №6 «Функции»

Введение. Повторяющиеся части кода целесообразно выносить в отдельные программные единицы – функции.

Обращение к функции осуществляется после ее задания. Функция может иметь несколько аргументов (указываются после имени функции в круглых скобках через запятую) или не иметь аргументов, может возвращать несколько результатов (после ключевого слова *return*) или не возвращать результат, но при этом изменять свои аргументы.

Возможна передача данных в функцию через глобальные переменные, которым должны быть присвоены значения до обращения к функции (однако, в общем случае так делать не рекомендуется). Аргументам функции могут придаваться значения по умолчанию – для случая обращения к функции без передачи фактических параметров этим аргументам.

Существуют также укороченные лямбда-функции (анонимные функции), в некоторых случаях более удобные в применении.

Пример 1. Программа для расчета значения функции двух переменных $f(x, y) = \sqrt[3]{x} + \sec(y^2)$.

```
from math import cos
print("")
print((-)*10+' Пример 1 '+'*10)
```

```
def f1(x):
    return x**a # a доступна как глобальная
```

```
def f2(y):
    return 1/cos(y**2)
```

```
def f1f2(x,y):
    return x**a, 1/cos(y**2)
```

```
a=1/3 # a - глобальная переменная
x=float(input('x=?'))
y=float(input('y=?'))
f=f1(x)+f2(y)
print('x=%11.5e' % (x), 'x=%11.5e' % (y), 'f(x,y)=%11.5e' % (f))
ff1,ff2=f1f2(x,y)
print('x=%11.5e' % (x), 'x=%11.5e' % (y), 'f(x,y)=%11.5e' % (ff1+ff2))
```

```
def f11(x,a): # a передается как аргумент
    return x**a
```

```
def f12(x,a=1/3): # a передается как аргумент со значением по умолчанию
    return x**a
```

```
ff = lambda x,y: x**a+1/cos(y**2) # анонимная функция возвращает только 1 значение!
```

```
a1=1/3
print('f1(x)=%11.5e' % (f1(x)), 'f11(x,a)=%11.5e' % (f11(x,a1)))
print('f12(x)=%11.5e' % (f12(x)))
print('ff(x,y)=%11.5e' % (ff(x,y)))
```

----- Пример 1 -----

x=?2

y=?3

x=2.00000e+00 x=3.00000e+00 f(x,y)=1.62383e-01

x=2.00000e+00 x=3.00000e+00 f(x,y)=1.62383e-01

f1(x)=1.25992e+00 f11(x,a)=1.25992e+00

f12(x)=1.25992e+00

ff(x,y)=1.62383e-01

Пример 2. Программа для расчета суммы N -мерных векторов \vec{a} и \vec{b} .

```
print("")
```

```
print((' ')*10+'Пример 2 ' + (' ')*10)
```

```
def Vecsum1(a,b,n):
```

```
    c=[0]*n
```

```
    for i in range(n): c[i]=a[i]+b[i]
```

```
    return c
```

```
def Vecsum2(a,b,n):
```

```
    return [ a[i]+b[i] for i in range(n) ]
```

```
def Vecsum3(a,b):
```

```
    return [ a+b for a,b in zip(a,b) ]
```

```
with open('data2.txt','r') as file1: # открытие файла на чтение
```

```
    lsta = file1.readline() # 1-я строка данных
```

```
    lstb = file1.readline() # 2-я строка данных
```

```
    a = [float(ai) for ai in lsta.split()] # список a
```

```
    b = [float(bi) for bi in lstb.split()] # список b
```

```
    print(' a=',a)
```

```
    print(' b=',b)
```

```
    n=len(a)
```

```
    c=Vecsum1(a,b,n)
```

```
    print(' c=',c)
```

```
    c=Vecsum2(a,b,n)
```

```
    print(' c=',c)
```

```
    c=Vecsum3(a,b)
```

```
    print(' c=',c)
```

----- data2.txt -----

1 -2 3

-4 5 -6

----- Пример 2 -----

```
a= [1.0, -2.0, 3.0]
```

```
b= [-4.0, 5.0, -6.0]
```

```
c= [-3.0, 3.0, -3.0]
```

```
c= [-3.0, 3.0, -3.0]
```

```
c= [-3.0, 3.0, -3.0]
```

Пример 3. Программа-калькулятор значений функций модуля *math* в командной строке.

```
print((-)*10+' Пример 3 '+(-)*10)
import math
print('Функции модуля math:\n',dir(math))
f=input('введите f(x):')
x=float(input('введите x:'))
y=eval('math.'+f)
print(' x=',x,f+'=',y)
```

----- Пример 3 -----

Функции модуля *math*:

```
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh',
'atan', 'atan2', 'atanh', 'ceil', 'comb', 'copysign', 'cos', 'cosh', 'degrees', 'dist', 'e', 'erf', 'erfc', 'exp',
'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose',
'isfinite', 'isinf', 'isnan', 'isqrt', 'lcm', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan',
'nextafter', 'perm', 'pi', 'pow', 'prod', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau',
'trunc', 'ulp']
```

введите f(x):cosh(2*x)

введите x:1

x= 1.0 cosh(2*x)= 3.7621956910836314

Задание. Написать программу и функцию для расчета заданной величины (без использования встроенных функций), провести сравнение с результатами Mathcad:

1. Экстремальные элементы матрицы $A = (a_{ij})_{i,j=1,\dots,N}$ и их индексы.
2. Произведение матрицы $A = (a_{ij})_{i,j=1,\dots,N}$ на столбец $\vec{x} = (x_1, \dots, x_N)^T$.
3. Номера треугольников, заданных своими вершинами $A_i(x_{Ai}, y_{Ai})$, $B_i(x_{Bi}, y_{Bi})$, $C_i(x_{Ci}, y_{Ci})$, $i = 1, 2, \dots, N$, и имеющих максимальную и минимальную площади.
4. Скалярное произведение N -мерных векторов \vec{x} и \vec{y} .
5. Норма $\|A\|_1 = \max_{1 \leq i, j \leq N} |a_{ij}|$ матрицы $A = (a_{ij})_{i,j=1,\dots,N}$.
6. Норма $\|A\|_2 = \sqrt{\sum_{i,j=1}^N a_{i,j}^2}$ матрицы $A = (a_{ij})_{i,j=1,\dots,N}$.
7. Комбинация $\alpha A^2 + \beta B$ матриц $A = (a_{ij})_{i,j=1,\dots,N}$ и $B = (b_{ij})_{i,j=1,\dots,N}$.
8. Определитель 3×3 -матрицы A .
9. Сумма $I_2(A) = a_{11}^2 + a_{22}^2 + \dots + a_{NN}^2$ для матрицы $A = (a_{ij})_{i,j=1,\dots,N}$.
10. Симметричная $\frac{1}{2}(A + A^T)$ и кососимметричная $\frac{1}{2}(A - A^T)$ составляющие квадратной матрицы A .
11. Матрица A^{-1} , обратная к 2×2 -матрице A .
12. Произведение AB матриц $A = (a_{ij})_{i,j=1,\dots,N}$ и $B = (b_{ij})_{i,j=1,\dots,N}$.
13. Номер столбца квадратной матрицы, сумма элементов которого максимальна, и номер строки, сумма элементов которой минимальна.
14. Значение первой производной полинома $P_n(x) = a_0 + a_1x^1 + \dots + a_nx^n$ в заданной точке $x = x_*$.
15. Матрица A^{-1} , обратная к 3×3 -матрице A .
16. Решение СЛАУ $A\vec{x} = \vec{b}$, где A – 2×2 -матрица.
17. Угол между трехмерными векторами \vec{x} и \vec{y} .

18. Произведение строки $\vec{x} = (x_1, \dots, x_N)$ на матрицу $A = (a_{ij})_{i,j=1,\dots,N}$.
19. Векторное произведение $\vec{x} \times \vec{y}$ трехмерных векторов \vec{x} и \vec{y} .
20. Собственные числа 2×2 -матрицы A .
21. Решение СЛАУ $A\vec{x} = \vec{b}$, где A – 3×3 -матрица.
22. Объем параллелепипеда, построенного на векторах \vec{a} , \vec{b} и \vec{c} .
23. Среднее квадратическое отклонение $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2}$ для выборки (y_1, \dots, y_N) ,
где \bar{y} – среднее арифметическое.
24. След $Tr(A) = a_{11} + a_{22} + \dots + a_{NN}$ матрицы $A = (a_{ij})_{i,j=1,\dots,N}$.
25. Норма вектора $\|\vec{x}\|_2 = \sqrt{|\vec{x}|}$.
26. Экстремальные элементы вектора $\vec{x} = (x_1, \dots, x_N)$ и их номера.
27. Количество инверсий в элементах вектора $\vec{x} = (x_1, \dots, x_N)$, т.е. таких ситуаций, что $x_i > x_{i+1}$.
28. Среднее геометрическое $\bar{y}_g = \sqrt[N]{y_1 \cdot y_2 \cdot \dots \cdot y_N}$ для выборки (y_1, \dots, y_N) .
29. Норма вектора $\|\vec{x}\|_1 = \max_{1 \leq i \leq N} |x_i|$.