

%% Лекция 6

%% Примеры. Функции работы с матрицами. Без циклов

%%

clc, clear

A=['r','o','p','o','d']

%%

b=[3,4,5]

c=[3,2,1]

d=[1,3,2,1]

%%

r1=A(b)

r2=A(c)

r3=A(d)

%%

B={r1,r2,r3}

%%

clc, clear

A=[1:10]

%%

b=A>5

%%

c=A(b)

%% the same

d=A(A>5)

%%

sum(d)

%% the same

A>5

A(A>5)

sum(A(A>5))

%%

A>5&A<8

%%

A(A>5&A<8)

%% the same

f=A(A>5&A<8)

%%

A>8|A<2

%%

A(A>8|A<2)

%%

g=A(A>8|A<2)

%%

g(1:2)

%%

clc, clear

a=[1:25]

%%

b=reshape(a,5,5)

%%

b'

%% fliplr flipud rot90

```

%% the same
A=(reshape([1:25],5,5))'
%%
B=A(1:2:end,:);
%%
C=A(:,2:2:end)
%%
D=C(1:2,1:2)
%%
E= repmat(D,2,2)
%% mod
mod(37,10)
%%
mod(4,2)==0
%%
mod(21,4)~=0
%%
F=mod(E,2)==0
%%
G=E.*F
spy(G,24,'<')
%%
clc
E
% any - хотя бы 1
any(E<4) % по столбцам
%%
any(any(E<4))
%%
p=magic(3)
m=mod(p,2)~=0
%%
% all - все
all(m)
%%
all(all(m))
%% the same
all(all(mod(p,2)~=0))
%%
all(all(p>0))
%%
sort(magic(5)) % по столбцам
%%
sort(magic(5),2) % по строкам
%%
sort(magic(3),2,'descend')
%%
ones(2)
rand(3)
tril(magic(5))
%%
spy(blkdiag(ones(1),rand(3),tril(magic(2))),16)
%%

```

```

H=eye(3)
find(H>0)
%%
J=round(9*rand(3)+1).*H
%%
K=nonzeros(J)
%%
L=diag(K)
M=L+fliplr(L)
%%
N= repmat(eye(2),3,3)
spy(N)
%%
P= repmat(M,2,2)
R=P(1:4,1:4)
%%
R(R==0)=32*sum(sum(N))
%%
R(R<256)=-1

```

%% Основы программирования в MATLAB

%% УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ

% Цикл for end

clc

```
for i=1:3
```

```
    i^2
```

```
end
```

%% С шагом

clc

```
for i=0:0.5:1
```

```
    i
```

```
end
```

%% Вложенные циклы

clc, clear

```
for i=1:3
```

```
    for j=i:3
```

```
        v(i)=i+j
```

```
    end
```

```
end
```

%%

clear, clc

```
for i=1:3
```

```
    for j=1:3
```

```
        A(i,j)=i^j;
```

```
    end
```

```
end
```

A

%% Цикл for end

clear, clc

```
A=[1 2 3 4]
```

```
for x=A
```

```
    x
```

```
end
```

```

%% Цикл for end
clear,clc
A=[1 2; 3 4]
for x=A
    x
end

%% Цикл while end
clear,clc
s=0
while s<5
    s=s+1
end
%%
clc
k=0;
while k<4
    k=k+1;
    if (k==2)
        continue
    else
        prod(1:k)
    end
end;

%% break
clc
k=0;
s=0;
while k<4
    k=k+1;
    r=round(10*rand(1)-5);
    if (r==0)
        break
    else
        s=s+r
    end
end;
%%

%% Множественный выбор switch end
clear,clc
x=5
switch x
    case {1,2,12}
        disp('winter')
    case {3,4,5}
        disp('spring')
    case {6,7,8}
        disp('summer')
    case {9,10,11}
        disp('autumn')
end

```

```

%% Операторы try и catch
n=randi(2,1)
A=randi(5,n)
b=randi(2,2,1)
try
    linsolve(A,b)
catch
    msgstr = lasterr
end

%%
% Включить таймер
t1=tic();
% Матрица гильберта 1000 порядка
h=hilb(1000);
% Вывод времени, прошедшего
%с момента запуска таймера
t2=toc(t1)
%%
clc
t1=tic();
x=rand(10,10);
x.^2;
t2=toc(t1)

%%
% М-файлы сценариев (script файлы) и функций должны иметь
уникальные имена.
% Длина имен не ограничивается, но только первый 31 символ
учитывается при
% идентификации имени.
% Если имя оказывается неуникальным,
% то соответствующий программный объект не исполняется,
% и выводится сообщение об ошибке.
%% Файлы-сценарии или файлы-скрипты. Пример
% Знак % в комментариях должен начинаться с первой позиции
строки.
clc;
clear;
[X, Y] = meshgrid([-3:0.1:3]);
Z = X.*exp(-X.^2-Y.^2);
surf(X, Y, Z)

%% Анонимные функции
% Анонимная функция – это однострочная функция MATLAB,
% основанная на выражении, для которой не требуется программный
файл.
sqr = @(n) n.^2;
x = sqr(3)
%%
y = 7;
fh = @(x)x.^2+y;
z = fh(2)

```

```

%% Структура M-файла-функции
% Имя файла совпадает с именем функции!
% function [var1,var2,...]=file_name(Список параметров)
% %Основной комментарий
% %Дополнительный комментарий
% Тело файла с любыми выражениями
% var1=выражение
% var2=выражение
% ...
% end
% Вызов функции:
% [var1,va2,...]=file_name(Список параметров)
%% Запуск файла-функции
[p,r]=stat([1,2,3,4,5,6])

%% Несколько функций в файле-функции
v = [1,2,3,4,5,6];
[m,s] = stat2(v)
[m,s] = stat2([1,2,3])

%% Несколько функций в файле-функции
myF(2,3)

%% О глобальных и локальных переменных
% Переменные в файлах-сценариях являются
% глобальными, а в файлах-функциях – локальными
% Команда
% global var1 var2...
% позволяет объявить переменные модуля-функции
% глобальными.

%% Функция error
clc
myF_error(2,0);
%%
clc
myF_error(2,2)

%% Функция nargchk
% Функция nargchk используется внутри m-файлов
% для проверки соответствия количества входных
% параметров
clc
msg = nargchk(4,9,5)

%% Функция lasterr
% Для вывода сообщения о последней произошедшей
% ошибке служит функция lasterr

```

%% Функции с переменным числом аргументов

```
% Вне тела m-файла функции
% nargin и nargout
% указывают, соответственно, количество входных
% или выходных аргументов, заданных
% пользователем.
```

```
clc
nargin(@myF)
nargout(@myF)
%%
nargin(@stat2)
nargout(@stat2)
```

```
%%
% Отрицательное число аргументов означает, что
% функция имеет переменное число аргументов;
% sort - переменное число входных аргументов
% eig - переменное число выходных аргументов
```

```
clc
clear
A=[1 2 3 ; 3 1 2 ; 2 3 9]
eig(A)
[V,D]=eig(A)
%%
nargin(@sort)
nargout(@eig)
```

```
%% Пример nargin
```

```
clc,clear
myPrint('Hello!')
myPrint('A',3.14,rand(1),magic(2))
```

```
%% Переменные varargin и varargout 19
```

```
clc
clear
A=rand(4,5,2)
[dim,rows,cols] = myDef(A)
%%
```

%% Файлы-функции в отдельных файлах!!!

%% stat.m

```
%%возвращает среднее значение и стандартное отклонение
```

```
function [m,s] = stat(x)
    n = length(x);
    m = sum(x)/n;
    s = sqrt(sum((x-m).^2/n));
end
```

%% stat2.m

```
function [m,s] = stat2(x)
    n = length(x);
    m = avg(x,n);
    s = sqrt(sum((x-m).^2/n));
end
```

```
function m = avg(x,n)
    m = sum(x)/n;
end
```

%% myF.m

```
function [ z ] = myF( x,y )
x=subF(x);
z=x+y;
end
function [ x ] = subF(x)
x=x^2;
end
```

%% myF_error.m

```
function [ z ] = myF_error( x,y )
if y==0
    z=0
    error(' Error - division by zero ');
else z=x/y;
end
```

%% myPrint.m

```
function [ ] = myPrint( varargin )
for i=1:nargin
    varargin(i)
end
```

%% myDef.m

```
function [s,varargout] = myDef(x)
nout = max(nargout,1) - 1;
% nargout - количество выходных аргументов
s = size(x);
for k=1:nout
    varargout{k} = s(k);
end
```