```matlab
%% Лекция 3. Matlab
% Любой объект в ML, в том числе скаляр, является массивом.
% Класс ARRAY является родительским для всех потомков.
%%
% Хранение массивов в ML осуществляется в векторной форме
% последовательно по столбцам, поэтому поддерживается как двойная
% (матричная) нумерация, так одинарная (векторная).
clc
clear
M=[1 2 3; 4 5 6; 7 8 9]
M(4)
M(4)==M(1,2)
%% Контроль типов
clc
clear
class(pi)
class('hello')
%%
clc
isa(pi, 'double')
isa('hello', 'char')
%%
clc
isnumeric(55)
islogical(1==2)
isreal(3.5)
%%
clc
isvector(1:5)
isempty([])
%%
clc
isinf(1/0)
isnan(0/0)
%%
clc
a=rand(2,3)
a(:,2)=0/0, isfinite(a)
%% Размер матрицы
clc
clear
A=eye(2,3)
numel(A) % 6 - общее число элементов
%% ndims - размерность
clc
ndims(17) % 2
ndims(1:5) % 2
ndims(zeros(2,3)) % 2
ndims(zeros(2,3,4)) % 3
%% size
clc
A=ones(2,6)
[m,n]=size(A)
%% length - длина большей размерности
length(A)
```

```
%% Методы класса Array
% ones / eye / zeros / magic
clc
eye(3)
%%
% linspace
clc
linspace(1,5)  % Вектор из 100 компонент
linspace(1,5,10)  % Вектор из 10 компонент
%% Заполнение матриц. reshape repmat
clear
clc
%A=reshape([1:12],3,4)
B=repmat([55, 44],1,2)
C=repmat([33, 22],2,1)
%% Заполнение матриц rand
clc
rand([2 3])
round(rand([2,3])*10)
round(rand([2,3])*10)-5
%% Заполнение матриц randi
clc
randi([-55,55],2,3)
randi(2,3) % 1:100
%%
clc
rand(1,5)
%%
% rand производит те же самые случайные числа
clc
rng('default') % Сбросить генератор случайных чисел,
rand(1,5)
rng('default')
rand(1,5)
%%
clc
s = rng
u1 = rand(1,5)
rng(s);
u2 = rand(1,5) % contains exactly the same values as u1
%%
clc
rng('shuffle'); % перемешать (достаточно 1 раза)
rand(1,5)
%% Удаление строк/столбцов
clear
clc
A=reshape([1:16],4,4)
A(:,3:end)=[]
A(3:end,:)=[]
%% Удаление строк/столбцов
clear
clc
A=eye(2)
B=repmat(A,4,4)
C=B(:,2:2:end), D=B(2:2:end,:)
```

```matlab
%% Объединение матриц cat
clear
clc
A=[1,2]
B=[3,4]
[A  B]
[A; B]
cat(1,A,B)
cat(2,A,B)
cat(3,A,B)
%% Изменение элементов матрицы
clear
clc
A=[1,2; 3,4]
A(1,1)=55
A(2,:)=123
%% Изменение элементов матрицы. прямоугольный фрагмент
clear
clc
A=ones(5)
A(3:end,2:4)=88
%% Функции min, max, sum, prod
clear
clc
A=[1024,-10; 3,4]
min(min(A))
min(A) % вектор минимальных
max(A) % вектор максимальных
%% sum, prod
clc
A=[1,2; 3,4]
sum(A)
sum(A,2)
%% abs
B=[1 -2; 3+i 4]
abs(B), abs(B(2,1))
%% fliplr / flipud / rot90
clc
A=[1,2; 3,4]
fliplr(A)
%%
clc
A=[1,2; 3,4]
flipud(A)
%% поворот против часовой стрелки
clc
A=[1,2; 3,4]
rot90(A)
%% diag
clc
A=[1,2; 3,4]
diag(A)
diag(A,1)
diag(A,-1)
diag(diag(A))
```

```
%% Методы класса Array. blkdiag
clc
m1=[1 2; 3 4]
m2=[10 20 30; 40 50 60]
m3=[2 4 6; 1 3 7; 5 4 3]
m=blkdiag(m1,m2,m3)
spy(m)
%% display – визуализация объектов, не подавляется знаком (;)
clc
display('Thinking vectorized');
disp([1 2]*[3 4]'); % 11
%% Логические операции с матрицами
clc
A=[1,2; 3,4]
A>=3
%% Логические операции с матрицами. Примеры
clear
clc
A=[10 2 10; 3 10 5]
A==10
sum(A==10)
sum(sum(A==10))
%% Не равно ~=
clear
clc
A=[1 2 1; 3 1 5]
B=[1 1 1; 6 7 8]
A~=B
%%
clear
clc
A=[1 2 1; 3 1 5]
B=[1 1 1; 6 7 8]
sum(sum(A==B))

%% Логическое И &
clear
clc
A=[1 2 1; 3 1 5]
B=[1 1 1; 6 7 8]
sum(sum((A==1)&(B==1)))

%% Логическое ИЛИ |
clear
clc
A=[1 2 1; 3 1 5]
B=[1 1 1; 6 7 8]
sum(sum((A==1)|(B==1)))

%% Сортировка - sort
clear
clc
A=[3,1,4,5,2]
B = sort(A) % сортирует по возрастанию
C = sort(A,'descend') % по убыванию
```

```
%% sort
clear
clc
A=[100 1 200 ; 500 77 400]
B = sort(A,2) % вдоль строк (по столбцам) по возрастанию
C = sort(A,2, 'descend') % вдоль строк (по столбцам) по убыванию

%% sort
clear
clc
A=[100 1 200 ; 500 77 400]
C = sort(A,2, 'descend') % вдоль строк (по столбцам) по убыванию

%% find поиск ненулевых элементов
clear
clc
A =[0 2*pi 0 ; 0 0 sqrt(pi); 0 0 pi/2]
[i, j, x] = find(A) % определяются индексы [i,j] и ненулевые %значения x матрицы A(i,j), записанные в вектор x
%% поиск элементов, удовлетворяющих заданному условию
clc
A
[i, j, x] = find(A>1.7) % вектор x состоит из логических единиц

%% Элементы линейной алгебры
%% Обратная матрица
% Если матрица А является квадратной и невырожденной,
% уравнения AX=E и XA=E (E – единичная матрица)
% имеют одинаковое решение X.
% Это решение называется матрицей обратной к А,
% и вычисляется с помощью функции inv (A) или A^(-1)
clc
clear
A= pascal(5) % матрица Паскаля
d = det (A)  % вычисляется определитель А
X = inv (A)  % вычисляется обратная матрица к А
Y = (A)^-1
X==Y
sum(sum(X==Y))

%% Собственные значения и собственные векторы матрицы
% Собственным значением и собственным вектором квадратной матрицы А
% называются скаляр λ и вектор v, удовлетворяющие условию Av=λv
clc
clear
A = [4 1 2; 3 7 1; 2 2 8];
lambda = eig(A)

%%
[V,D] = eig(A)
(A*V(:,1)-D(1,1)*V(:,1))<1.e-13
(A*V(:,2)-D(2,2)*V(:,2))<1.e-13
(A*V(:,3)-D(3,3)*V(:,3))<1.e-13
% собственные значения на главной диагонали диагональной матрицы D
% собственные векторы в матрице V записаны по столбцам
```

```
%% Норма и число обусловленности матрицы
clear
A= rand(1,4)
norm(A)
%% cond число обусловленности – характеризует устойчивость решения системы
B=A+A'
cond(B)

%% Решение СЛАУ методом Гаусса Ax=b
clc
clear
A = [4 1 2;3 7 1;2 2 8];
b = [7; 11; 12];
x = A\b
A*x-b

%% функция mldivide равносильна операции деления (\)
x = mldivide(A, b)

%% Проверка точности решения
b - A*x % невязка
norm(b - A*x) %норма вектора невязки
%% Решение СЛАУ с помощью процедуры linsolve
% help linsolve
clc
clear
A=[4 1 2;3 7 1;2 2 8];
b = [7; 11; 12];
x=linsolve(A,b)

%% Факторизация матриц
% LU-разложение матрицы
% https://docs.exponenta.ru/matlab/ref/lu.html
% факторизация A на
% произведение l – нижней треугольной и u – верхней треугольной матриц,
% такая что l*u=A
clc
clear
A = [4 1 2;3 7 1;2 2 8]
[l,u]=lu(A)
l*u==A
%%

% Разложение QR
% https://docs.exponenta.ru/matlab/ref/qr.html
% факторизация A на
% R – верхнюю треугольную и
% Q – унитарную (в вещественном случае ортогональную (Q*Q'=E))
% матрицы такие, что Q*R=A
%%

% Факторизация Холецкого
% https://docs.exponenta.ru/matlab/ref/chol.html
%%
```