MatLab 3

Дербенев Леонид

Грохотова Елена

M-файлы

M-файлы являются обычными текстовыми файлами, которые создаются с помощью текстового редактора.

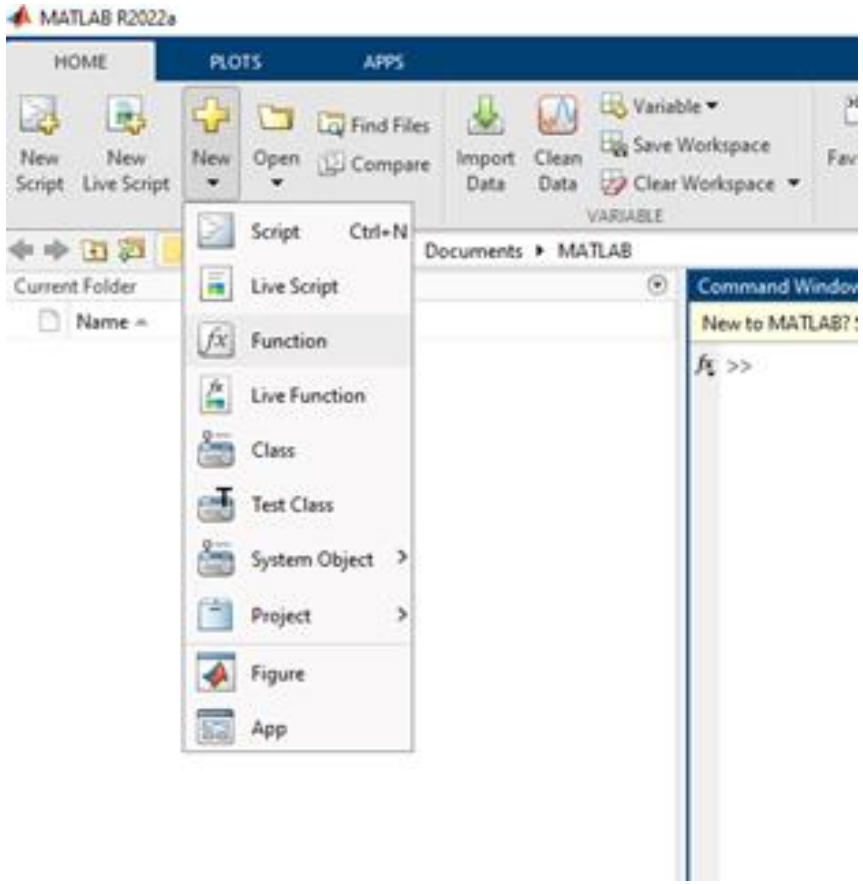
В языке Matlab имеются программы двух типов с расширением *.m.

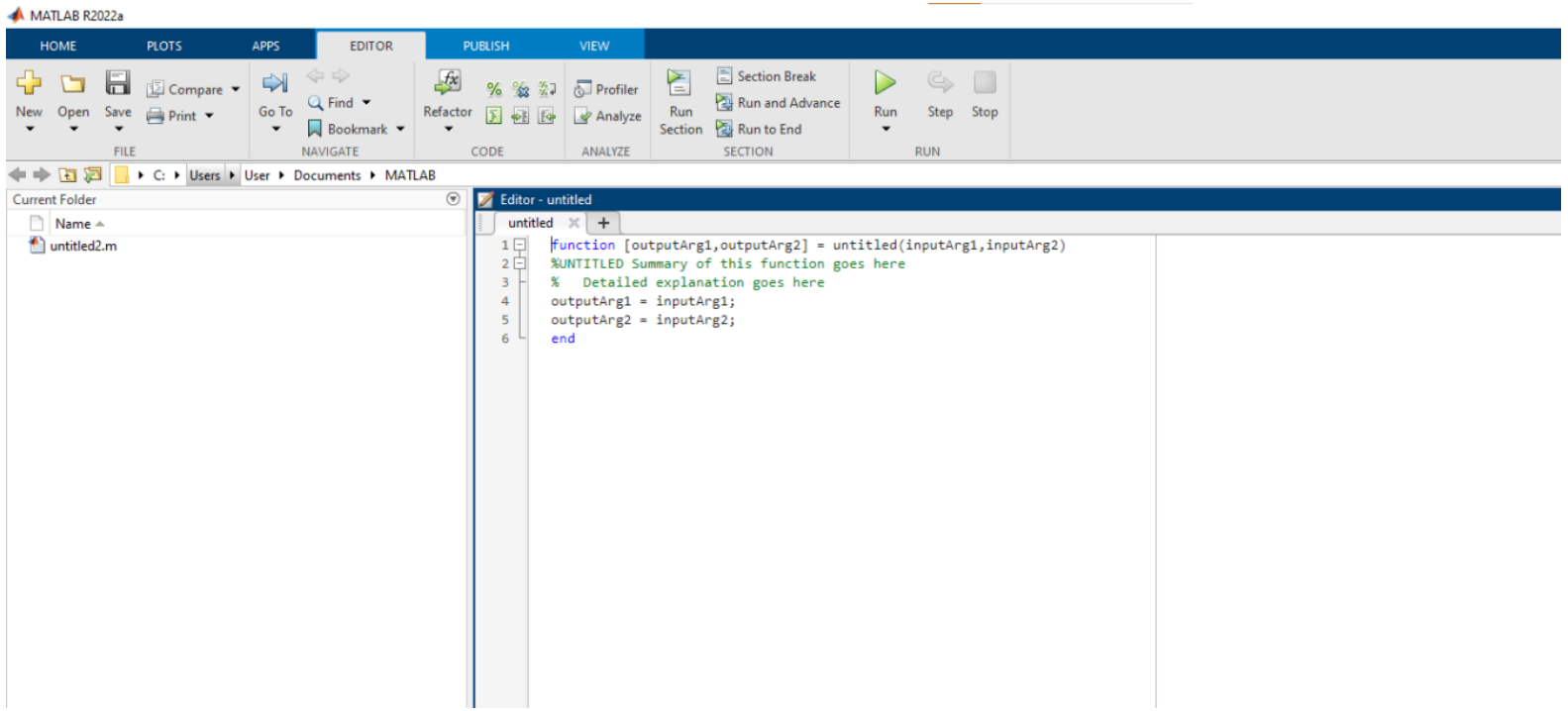
· Файл-функции (процедуры).

· Script-файлы(файлы-сценарии или управляющие программы)

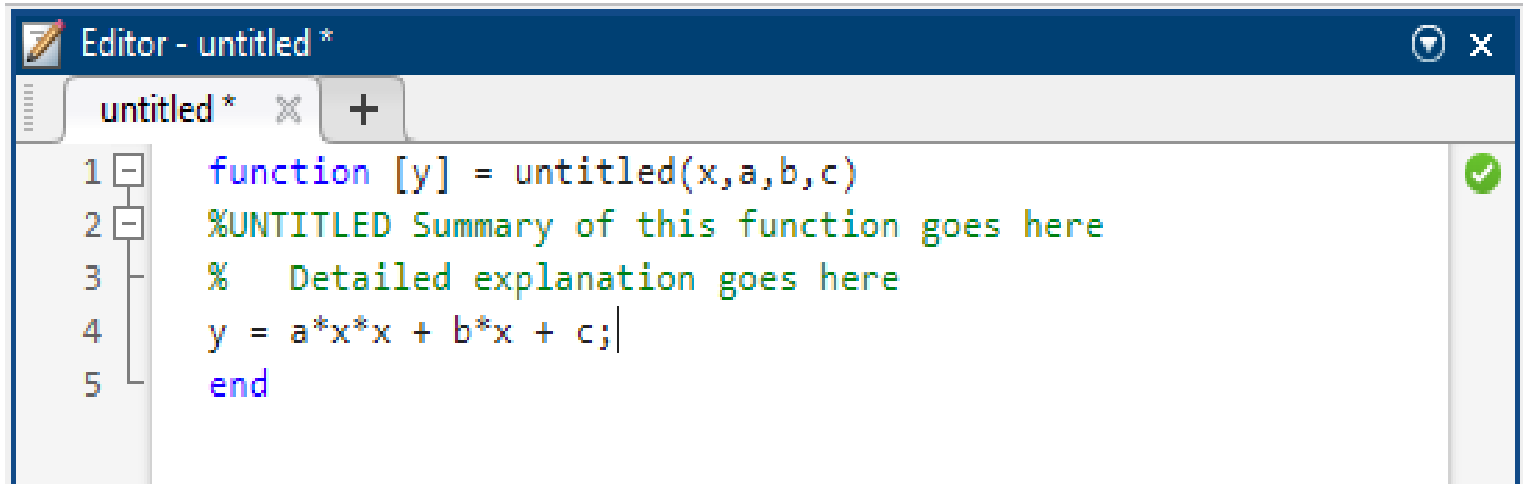
Файл-функции

Файл-функции оформляются отдельные процедуры и функции т.е. такие части программы, которые рассчитаны на неоднократное использование Script-файлами или другими процедурами при изменяемых значениях входных параметров и не могут быть выполнены без предварительного задания значений переменных, которые называют входными.


MATLAB R2022a
HOME
PLOTS
APPS
New Script
New Live Script
New
Open
Find Files
Compare
Import Data
Clean Data
Variable
Save Workspace
Clear Workspace
VARIABLE
Documents ▸ MATLAB
Current Folder
Name
Command Window
New to MATLAB?
>>
Script Ctrl+N
Live Script
Function
Live Function
Class
Test Class
System Object
Project
Figure
App

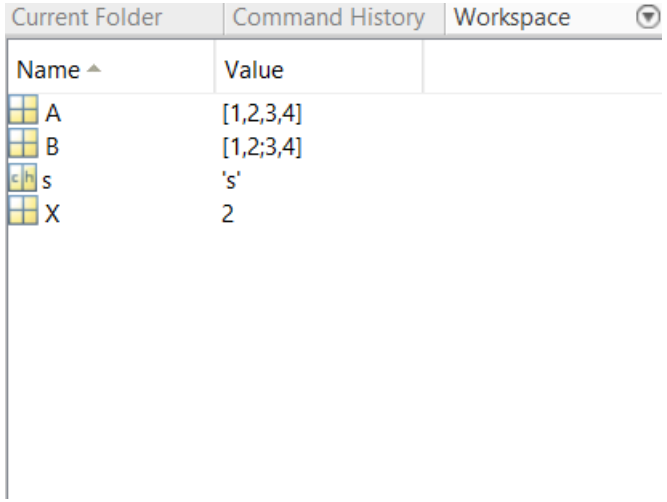

MATLAB R2022a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
New
Open
Save
Compare
Print
FILE
Go To
Find
Bookmark
NAVIGATE
Refactor
CODE
Profiler
Analyze
ANALYZE
Run Section
Section Break
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
C: ▸ Users ▸ User ▸ Documents ▸ MATLAB
Current Folder
Name
untitled2.m
Editor - untitled
untitled
function [outputArg1,outputArg2] = untitled(inputArg1,inputArg2)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
outputArg1 = inputArg1;
outputArg2 = inputArg2;
end

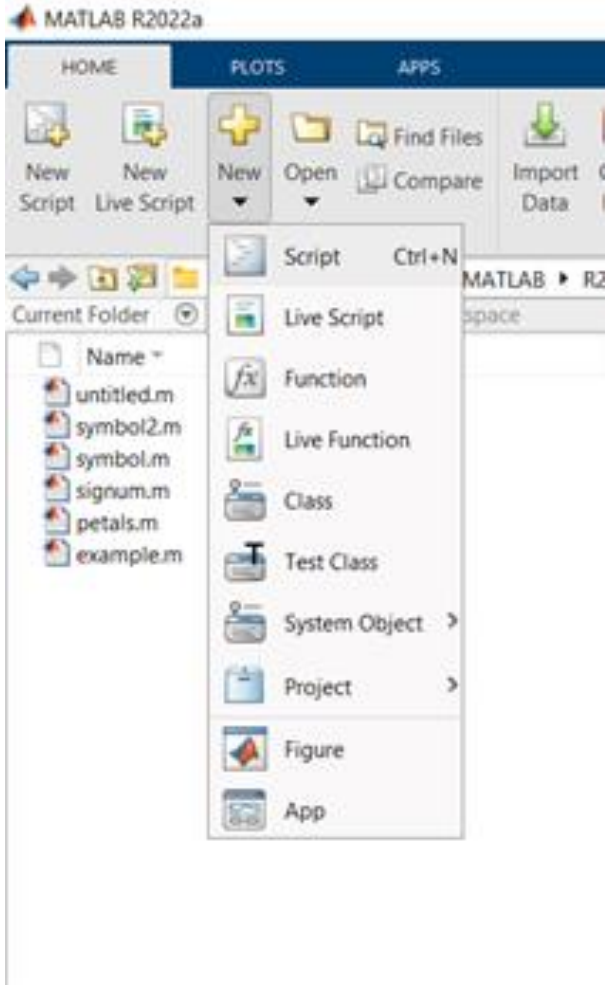
- function [перечень выходных величин] = имя процедуры (перечень входных величин).

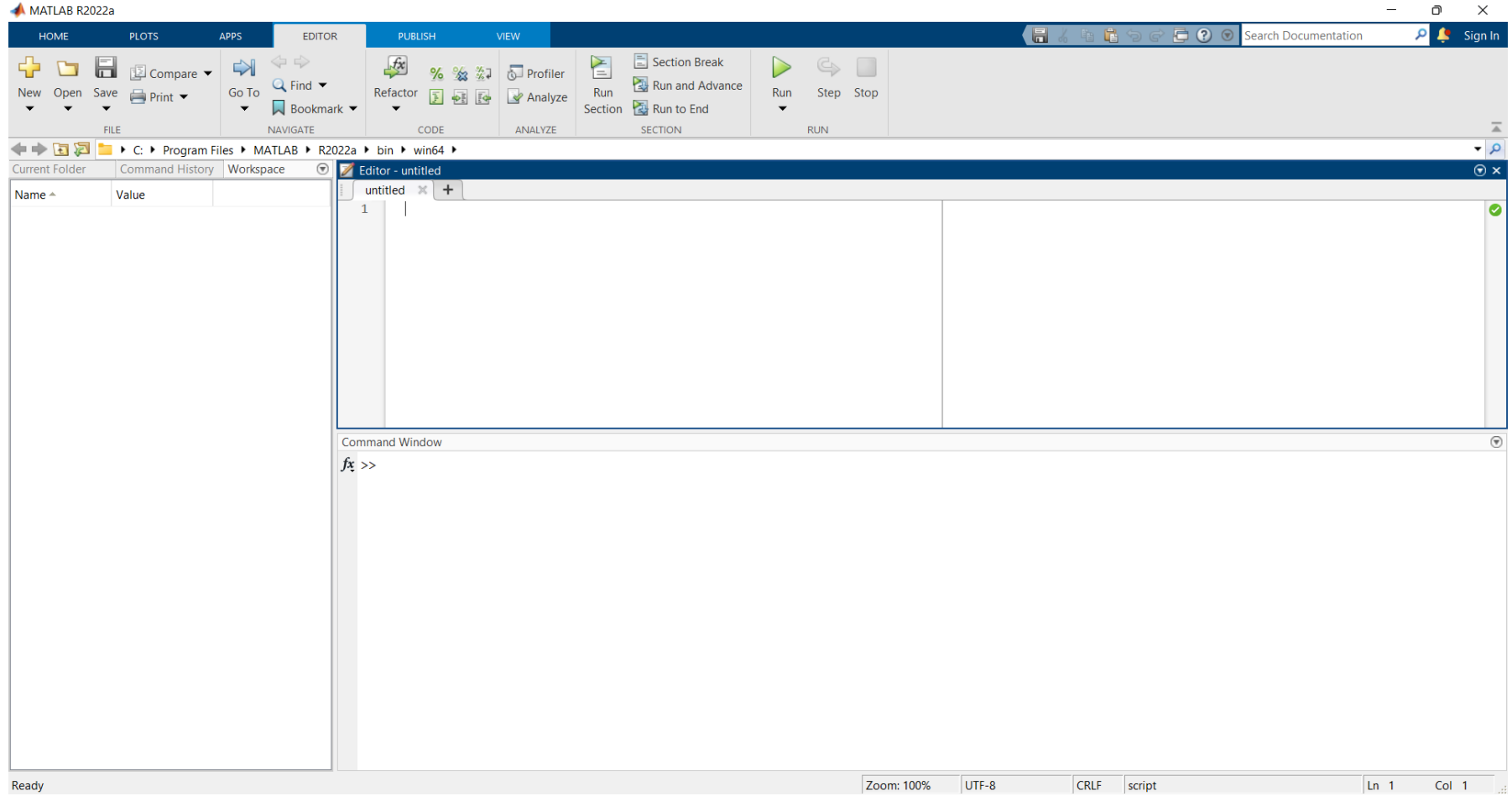


Файлы-сценарии (Script-файлы)

- Файлы-сценарии представляет просто последовательность команд MATLAB без входных и выходных параметров.
- В файлах-сценариях все используемые переменные образуют рабочее пространство (Work Space).




MATLAB R2022a
HOME
PLOTS
APPS
New Script
New Live Script
New
Open
Find Files
Compare
Import Data
Current Folder
Name
untitled.m
symbol2.m
symbol.m
signum.m
petals.m
example.m
Script Ctrl+N
Live Script
Function
Live Function
Class
Test Class
System Object
Project
Figure
App


MATLAB R2022a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Compare
Print
Go To
Find
Bookmark
Refactor
Profiler
Analyze
Run Section
Section Break
Run and Advance
Run to End
Run
Step
Stop
FILE
NAVIGATE
CODE
ANALYZE
SECTION
RUN
C: Program Files MATLAB R2022a bin win64
Current Folder
Command History
Workspace
Name
Value
Editor - untitled
untitled
1
Command Window
>>
Ready
Zoom: 100%
UTF-8
CRLF
script
Ln 1
Col 1

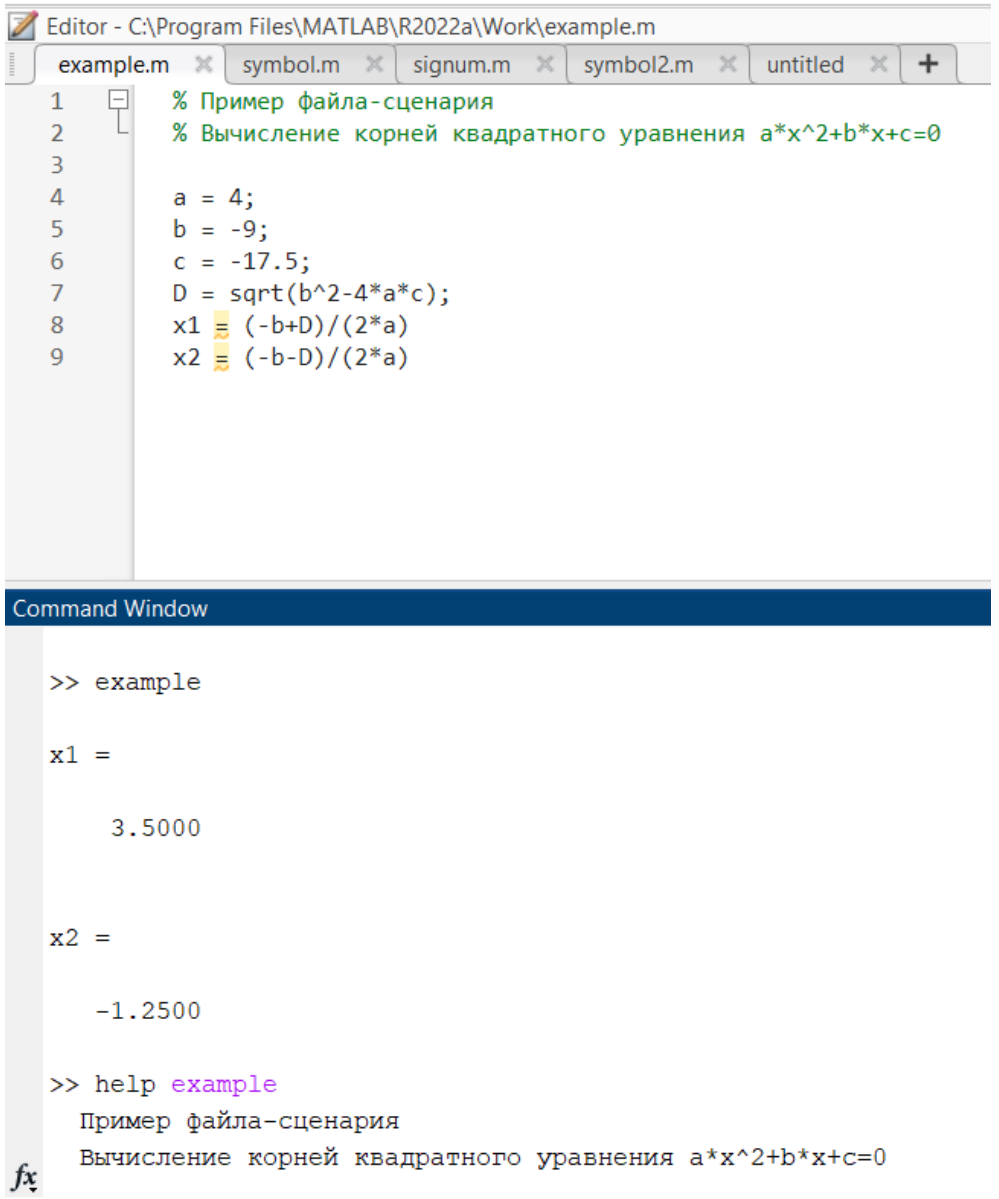
- Несколько операторов в одной строке разделяются символами “ ; ” или “ , ”. Пробел является разделителем только элементов массива внутри квадратных скобок.
- Если оператор не заканчивается символом “ ; ”, то результат его действия при выполнении программы будет выведен в командное окно.
- Длинный оператор можно записать в несколько строк, используя знак переноса - три точки (…).

```
>> m=1

m =

     1

>> m=1;n=m+1

n =

     2
```

```
>> m=1+2+3 + 4 + 5 + ...
6 + 7 + 8 + 9

m =

    45
```

- Строка программы или её часть, начинающаяся с символа “ % ” не выполняется, она воспринимается системой как комментарий.

- Строки комментария, предшествующие первому выполняемому оператору, воспринимаются как описание программы и выводятся в командное окно по команде help<имя файла>.



- Операторы начала и окончания текста программы отсутствуют, т.е. начало и конец программы никак не маркируются.
- Имена переменных могут содержать лишь буквы латинского алфавита или цифры и должны начинаться с буквы. Общее число символов - не более 19.
- В именах переменных могут использоваться как прописные, так и строчные буквы с учетом того, что система Matlab их различает.

Операторы ветвления

if Условие, Действие, end

if <выражение1>

<операторы1>

elseif <выражение2>

<операторы2>

….

else

<операторы3>

end

Для реализации составных условий в MatLab используются логические операторы:

& - логическое И
| - логическое ИЛИ
~ - логическое НЕ

Пример

```
x = 1;
y = 50;
if ((x >= 0.9) && (y >= 60))
disp(‘ok’);
else disp(‘not right’)
end
```

```
switch switch_expr
 case case_expr
    statement, ..., statement
  case {case_expr1, case_expr2, case_expr3, ...}
    statement, ..., statement
  otherwise
    statement, ..., statement
end
```

Циклы в MatLab

Цикл for

for переменная = значения
    действие
end

```
Command Window
>> for m = 1:4
       m
   end

m =

     1


m =

     2


m =

     3


m =

     4

fx >>
```

В этом цикле выводится значение m.

```
ok.m | cyc.m
1  for s = 1.0: -0.1: 0.0
2     disp(s);
3  end
```

```
Command Window
>> cyc
    1

    0.9000

    0.8000

    0.7000

    0.6000

    0.5000

    0.4000

    0.3000

    0.2000

    0.1000

         0

fx >>
```

Цикл **for** идёт от 1 до 0 с шагом -0.1.

```
Editor - C:\Users\User\Documents\MATLAB\untitled2.m
untitled | untitled2.m
1  A = [1 , 3, 4; 1, 2, 3]
2  for i = A
3      disp(i);
4  end
```

```
Command Window
New to MATLAB? See resources for Getting Started.
>> untitled2

A =

     1     3     4
     1     2     3

     1
     1

     3
     2

     4
     3

>> for s = [1,5,8,17]
   disp(s)
end
     1

     5

     8

    17
```

Переменная **s** будет последовательно приравниваться 1, 5 ,8 , 17 и выводиться.

Цикл While

```
while  Условие
       Действие
end
```

Также в условии **while** можно использовать логические операторы
И — && и ИЛИ — ||, записывая несколько логических выражений в условие.

```
>> eps = 5;
while eps > 1
    eps = eps - 1
end

eps =

     4


eps =

     3


eps =

     2


eps =

     1
```

```
>> S = 0;                    % начальное значение суммы
i=1;                         % счетчик суммы
while i <= 20                % цикл (работает пока i<=10
     S=S+i;                  % подсчитывается сумма
     i=i+1;                  % увеличивается счетчик на 1
    if S > 20                    % если S > 20,
         break;                   % то цикл завершается
     end
end                          % конец цикла
disp(S);                     % отображение суммы 21 на экране
    21
```

```
>> S = 0;                      % начальное значение суммы
a = [1 2 3 4 5 6 7 8 9]; % массив
i=0;                         % счетчик индексов массива
while i < length(a)          % цикл (работает пока i меньше
                             % длины массива a)
     i=i+1;                  % увеличивается счетчик индексов на 1
    if i == 5                    % если индекс равен 5
         continue;           % то его не подсчитываем
     end
     S=S+a(i);               % подсчитывается сумма элементов
end                          % конец цикла
disp(S);                     % отображение суммы 40 на экране
    40
```

Load

Функция **load** позволяет загрузить из указанного файла ранее сохраненные переменные.

Имя файла, заключенное в апострофы, указывается во входном аргументе load:

V = load(‘имя_файла’);

Файл с данными должен находится в текущем каталоге MatLab, иначе требуется указать полное имя файла.

vec0.txt – Блокнот

Файл Изменить Просмотр

```
1
2
3
-13.5
3478
0.875
11
```

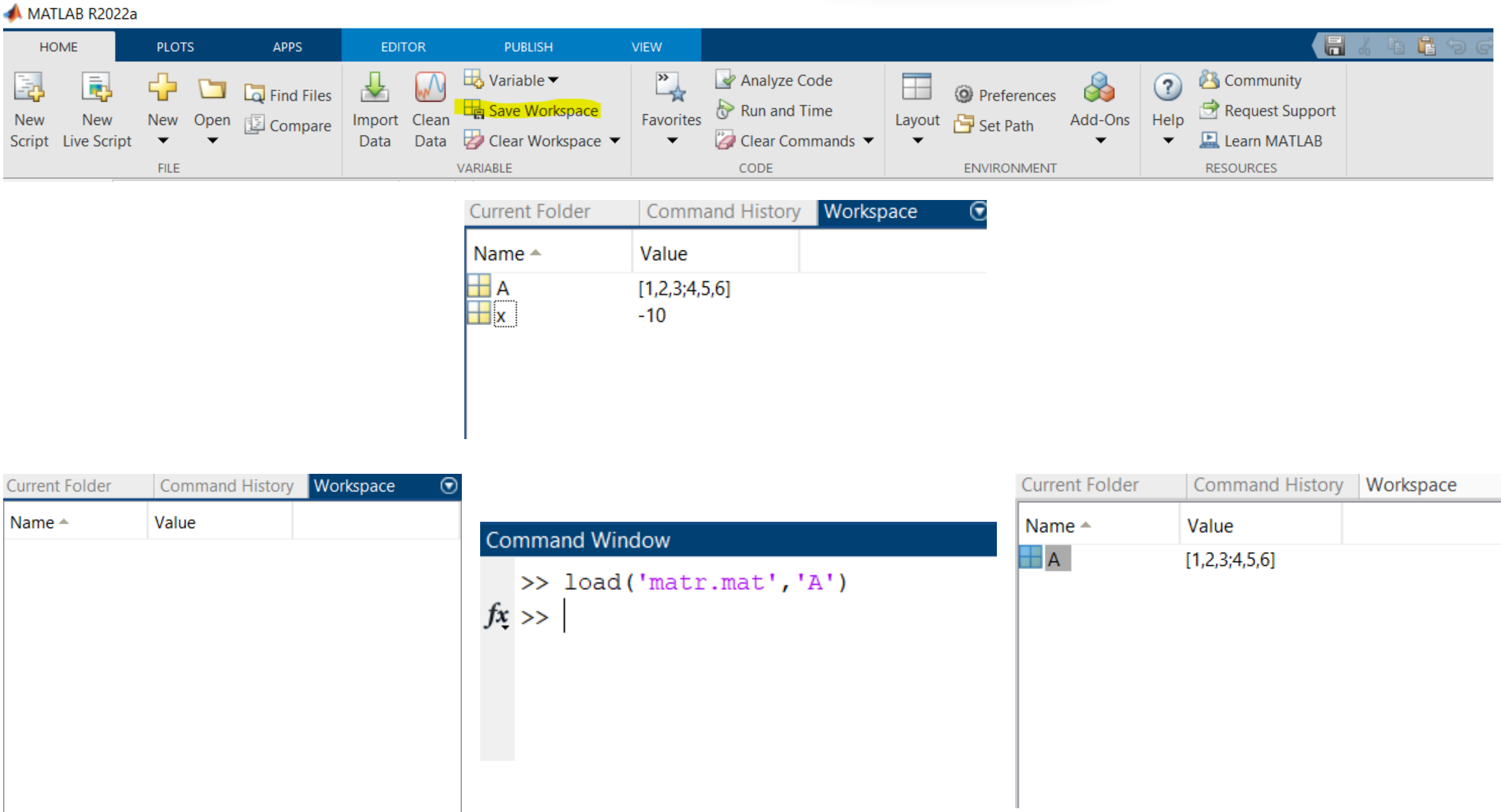
vec1.txt – Блокнот

Файл Изменить Просмотр

```
1 2 3 4
```

Command Window

```
>> v0=load('vec0.txt')

v0 =

   1.0e+03 *

    0.0010
    0.0020
    0.0030
   -0.0135
    3.4780
    0.0009
    0.0110

>> v1=load('vec1.txt')

v1 =

     1     2     3     4

>>
```

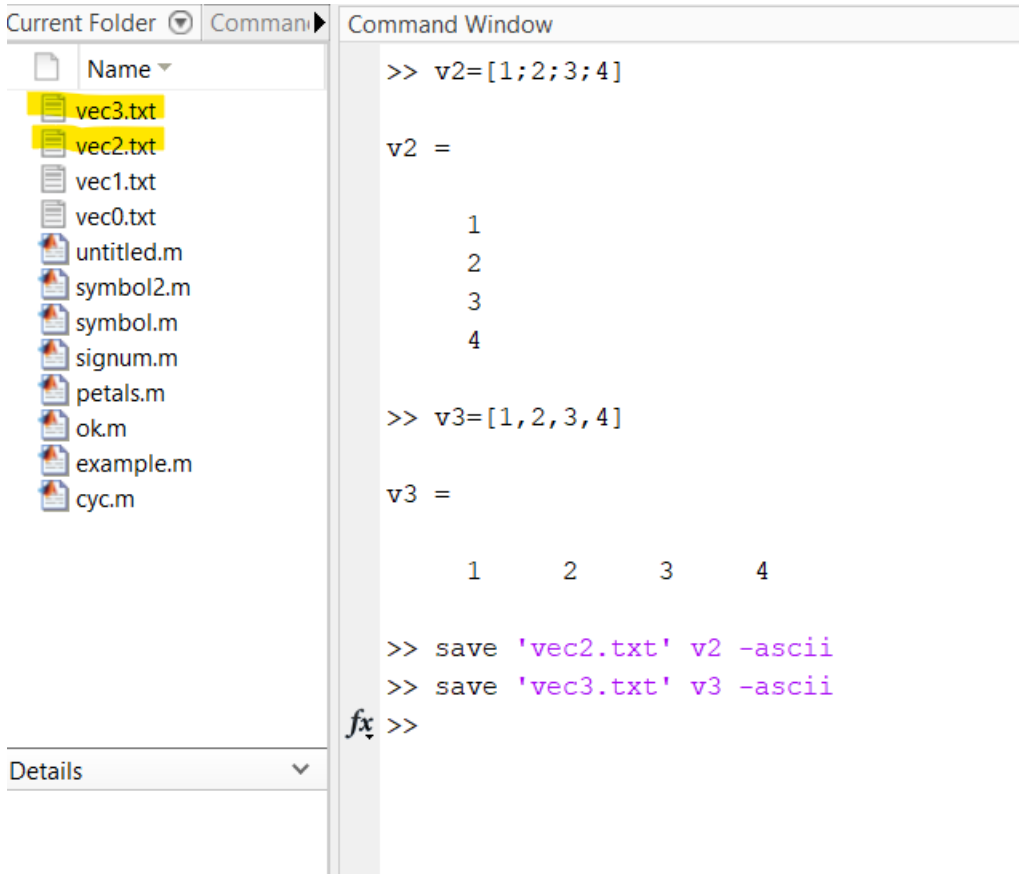

MATLAB R2022a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
New Script
New Live Script
New
Open
Find Files
Compare
Import Data
Clean Data
Variable
Save Workspace
Clear Workspace
Favorites
Analyze Code
Run and Time
Clear Commands
Layout
Preferences
Set Path
Add-Ons
Help
Community
Request Support
Learn MATLAB
FILE
VARIABLE
CODE
ENVIRONMENT
RESOURCES
Current Folder
Command History
Workspace
Name
Value
A
[1,2,3;4,5,6]
x
-10
Current Folder
Command History
Workspace
Name
Value
Command Window
>> load('matr.mat','A')
>>
Current Folder
Command History
Workspace
Name
Value
A
[1,2,3;4,5,6]

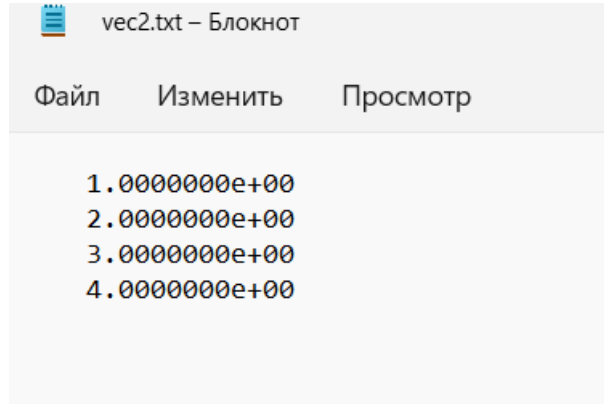
Save

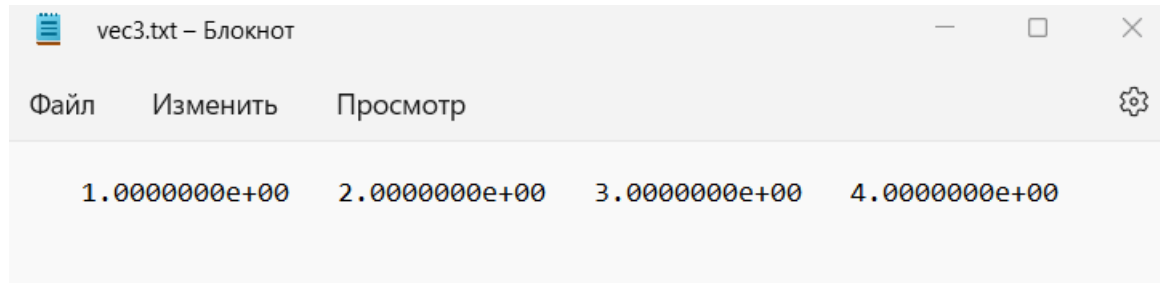
Функция **save** позволяет сохранять (записывать) произвольные переменные программы в файл.

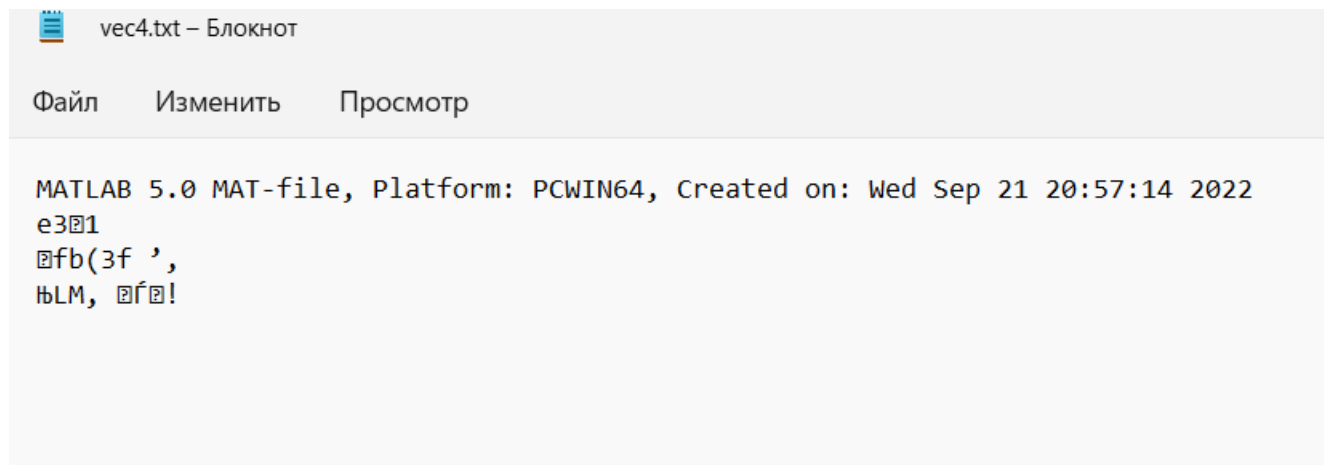
Её аргументами являются: имя файла, переменная, и дополнительный параметр `-ascii`, означающий запись в текстовом виде:

```
save 'vec2.dat' v2 -ascii
```


Current Folder
Command
Name
vec3.txt
vec2.txt
vec1.txt
vec0.txt
untitled.m
symbol2.m
symbol.m
signum.m
petals.m
ok.m
example.m
cyc.m
Details
Command Window
>> v2=[1;2;3;4]
v2 =
1
2
3
4
>> v3=[1,2,3,4]
v3 =
1 2 3 4
>> save 'vec2.txt' v2 -ascii
>> save 'vec3.txt' v3 -ascii
>>


vec2.txt – Блокнот
Файл Изменить Просмотр
1.0000000e+00
2.0000000e+00
3.0000000e+00
4.0000000e+00


vec3.txt – Блокнот
Файл Изменить Просмотр
1.0000000e+00 2.0000000e+00 3.0000000e+00 4.0000000e+00


vec4.txt – Блокнот
Файл Изменить Просмотр
MATLAB 5.0 MAT-file, Platform: PCWIN64, Created on: Wed Sep 21 20:57:14 2022
e3□1
□fb(3f ’,
ЊLM, □Ґ□!

```
>> v=1

v =

     1

>> save 'matr1.mat' v
```

Считывание, запись матриц

Command Window

```
>> B=[1,2,3;4,5,6;7,8,9]

B =

     1     2     3
     4     5     6
     7     8     9

>> save 'matrix1.txt' B -ascii
>>
```

matrix1.txt – Блокнот

Файл Изменить Просмотр

```
1.0000000e+00   2.0000000e+00   3.0000000e+00
4.0000000e+00   5.0000000e+00   6.0000000e+00
7.0000000e+00   8.0000000e+00   9.0000000e+00
```

matrix0.txt – Блокнот

Файл Изменить Прос

```
0 1 2 3 4
0 0 0 0 0
1 0 2 0 1
1 1 1 1 1
```

Command Window

```
>> A=load('matrix0.txt')

A =

     0     1     2     3     4
     0     0     0     0     0
     1     0     2     0     1
     1     1     1     1     1

>>
```

fwrite, fread

Для чтения бинарных файлов (последовательность произвольных байтов, бинарным он называется потому, что все записи внутри файла делаются только при помощи «1» и «0») разработаны:

- fwrite(<идентификатор файла>, <переменная>, <тип данных>);

и

- A = fread (<идентификатор файла>) считывает данные в двоичном формате из файла в матрицу A

  A = fread (<идентификатор файла>, count) считывает количество элементов, заданное параметром count.

<идентификатор файла> - это указатель на файл. Чтобы получить идентификатор, используется функция

<идентификатор файла> = fopen(<имя файла>,<режим работы>);

fopen служит для открытия файла и вызывается с двумя входными аргументами: именем файла и строкой, задающей способ доступа к файлу (режим работы).

Выходным аргументом fopen является идентификатор файла. Функция fopen возвращает –1, если при открытии файла возникла ошибка.

Существует четыре основных способов открытия файла:

1) `f=fopen('myfile.dat','rt')` — открытие текстового файла myfile.dat только для чтения из него;
2) `f=fopen('myfile.dat','rt+')` — открытие текстового файла myfile.dat для чтения и записи данных;
3) `f=fopen('myfile.dat','wt')` — создание пустого текстового файла myfile.dat только для записи данных;
4) `f=fopen('myfile.dat','wt+')` — создание пустого текстового файла myfile.dat для записи и чтения данных.

По окончании работы с файлом необходимо закрыть его при помощи `fclose(f)`.

Описанные выше функции работы с файлами (fread, fwrite) позволяют записывать и считывать информацию по байтам. Иногда считывать по байтам затруднительно. Поэтому для этих целей были специально разработаны функции чтения.

A = fscanf(идентификатор файла, 'формат', число считываемых элементов)

(форматная строка может состоять из спецификаторов)

и записи

fprintf(идентификатор файла, 'форматы', список переменных)

Работа с текстовыми файлами состоит из трех этапов:

1) открытие файла;
2) считывание или запись данных;
3) закрытие файла.

Список основных спецификаторов для fscanf(), fprintf()

Спецификатор	Описание
%d	целочисленные значения
%f	вещественные значения
%s	строковые данные
%c	символьные данные
%u	беззнаковые целые значения

```
1  174,1.63,2.22
2  180,1.34,1.09
3  181,2.63,1.63
4  183,1.63,55.64
```

file.m

```
1  fid = fopen('my_file.dat', 'rt');
2
3  S = fscanf(fid, '%d,%f,%f',[3,3]);
4  fclose(fid);
```

Command Window

```
>> file
>> S

S =

  174.0000  180.0000  181.0000
    1.6300    1.3400    2.6300
    2.2200    1.0900    1.6300

fx >>
```

Размеры формируемого массива указываются в векторе в третьем входном аргументе fscanf.

- Для записи данных в текстовый файл в заданном формате используется функция fprintf().

file.m

```
1  fid = fopen('my_file.dat', 'wt');
2
3  fprintf(fid, '%3d;%d;%.4f\n', X');
4  fclose(fid);
```

Command Window

```
>> X=[1,2,3.45;4,5,6.23]

X =

    1.0000    2.0000    3.4500
    4.0000    5.0000    6.2300

>> file
```

```
1  1;2;3.4500
2  4;5;6.2300
3
```

спецификатор %6d говорит о том, что целые числа должны иметь 6 значащих цифр, а спецификатор %.4f означает, что после запятой будет отображено только 4 цифры.

file.m

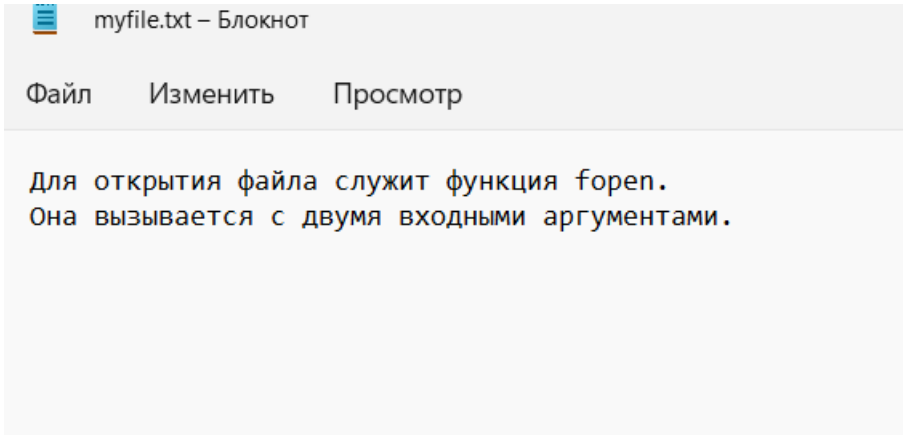
```
1  fid = fopen('my_file.dat', 'wt');
2
3  fprintf(fid,'%s%f%s%f','x=',x,' y=',x^2);
4  fclose(fid);
```

Command Window

```
>> x=2.4

x =

    2.4000

>> file
>> file
>>
```

```
1  x=2.400000 y=5.760000
```

Построчное считывание информации из текстового файла производится при помощи функции `fgetl`. Входным аргументом `fgetl` является идентификатор файла, а выходным — текущая строка. Каждый вызов `fgetl` приводит к считыванию одной строки и переводу текущей позиции в файле на начало следующей строки.




```
>> f=fopen('myfile.txt','rt');
str1=fgetl(f);
str2=fgetl(f);
str3=fgetl(f);
fclose(f);
>> str1

str1 =

    'Для открытия файла служит функция fopen.'

>> str2

str2 =

    'Она вызывается с двумя входными аргументами.'

>> str3

str3 =

    -1
```