

Лабораторная работа 1

Работа в командном окне и (или) в текстовом редакторе

(используйте отладчик для интерактивной реализации кода)

Системные переменные

eps – бесконечно малое, обусловленное точностью (2.2204e-016)

realmax – наибольшее положительное вещественное число (1.7977e+308)

realmin – наименьшее положительное вещественное число (2.2251e-308)

intmax – наибольшее положительное целое число (2147483647)

intmin – наименьшее положительное целое число (-2147483648)

ans – системная переменная (результат последней операции, в отсутствии результирующей переменной)

pi – 3.141592653589793 (format long)

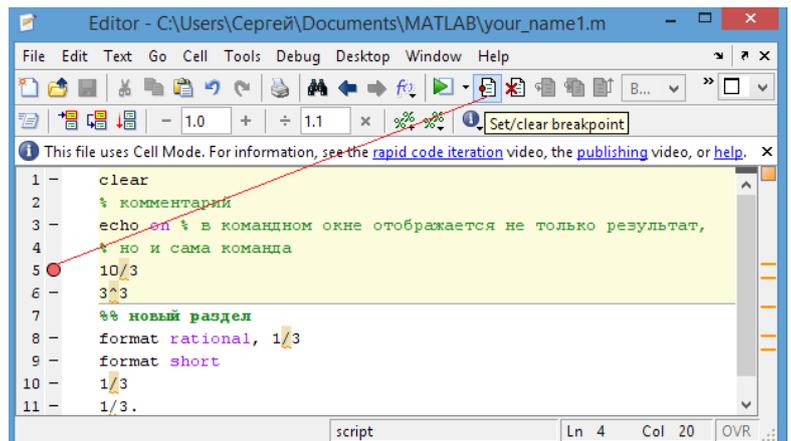
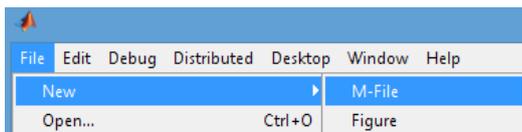
inf – бесконечно большое

NaN (not a number) – нечисловой формат

exp(1) (2.718281828459046) нет зарезервированной системной переменной

end – последний элемент

Простейшие операции



Задание 1. Выполните вычисления:

1+2

3-4

3*5

10/3

3^3

format rational

1/3

format short

1/3, 1/3.

1/0 % результат - системная переменная

0/0 % нечисловой формат - системная переменная

Задание 2

Задайте векторы:

d1=[1 1 1], d2=[2 2 2]

d3=[5; 5; 5], d4=d2'

Выполните операции

d1+d2, d4+d3,

d1*d2, d4*d2,

d1/d2, d2/d1

(объясните результат)

Замечание. Воспользуйтесь стеком (по стрелке вниз-вверх на клавиатуре) для выбора и редактирования исполненной ранее строки, если это целесообразно.

Задание 3

Выполняйте и анализируйте ∇ результат

```
a='MatLab forever'
```

(результаты заданий 1-2 остаются в сессии)

```
who
```

whos % почему в **a** такое количество байт?

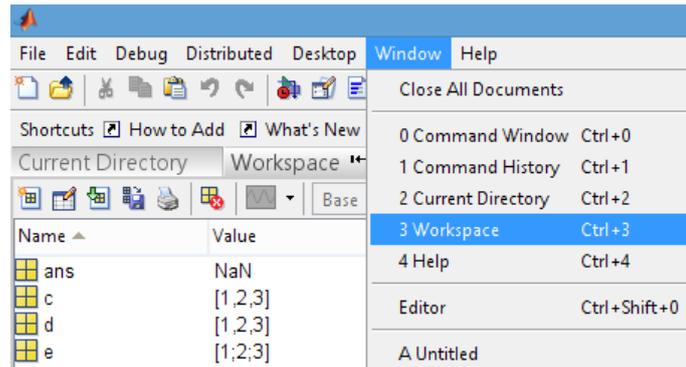
```
clear d2
```

```
whos
```

clear d1,d3 % объясните результат

```
who
```

```
clear
```



Панель Workspace – содержит информация о глобальных переменных (команда **who** – интерактивный аналог); **whos** - расширенная информация о данных (+ какие) типы данных (классы)

Конструкторы матриц и операции с векторами и матрицами

Задание 4. Матрицы из нулей (конструктор **zeros**)

Выполните команды:

```
zeros(5) % создается в памяти матрица пятого порядка из нулей
```

```
zeros(2,3)
```

```
zeros(4,3,5)
```

```
A=[1,2,3;4,5,6]
```

```
B= zeros(size(A))
```

2

Задание 5. Матрицы из единиц (**ones**) и единичная матрица (**eye**)

Выполните команды:

```
ones(5)
```

```
ones(2,3)
```

```
ones(4,3,5)
```

```
A=[1,2,3;4,5,6]
```

```
eye(5) % единичная матрица
```

```
B=eye(size(A))
```

Задание 6. Случайные матрицы

rand(5) % матрица 5-го порядка, заполненная случайными числами, равномерно распределенными на интервале (0, 1)

randn(10,2) % матрица заданного порядка, случайные числа генерируются с помощью нормального распределения нулевым средним и единичной дисперсией

randi([3 9],3,3) % генерируется случайная матрица третьего порядка, элементами которой являются целые числа от 3 до 9 включительно. (Читайте справочную систему **help randi**)

Задание 7. Выбор элементов матрицы, редактирование

Задайте матрицы:

A=[1 2 3; 4 5 6; 7 8 9]; % заметим, что ; - подавляет вывод результата

B=[1 1 1; 2 2 2; 3 3 3];

Для доступа к элементам массива используются круглые скобки A(2,3) – элемент 2-й строки и 3-го столбца; B(1:3) - выбран столбец [1;2;3], B(2,:) – вторая строка
Выберите 1-й столбец матрицы A и 2-й столбец матрицы B.
Выберите вторую и третью строку матриц.

Замечание. В памяти Matlab любая матрица хранится как вектор-строка, построенная последовательно из её векторов-столбцов. Наличие одинарной нумерации предполагает такое соответствие.

A=rand (6)

AL=tril(A), tril(A,-2)

AU= triu(A), triu(A,2)

AD=diag(A), v=diag(A,3),

B=diag(v), diag(A),

Факультативно:

fliplr(A) – поворот матриц вправо-влево

flipud(A) – поворот матриц вверх-вниз

rot90(A) – поворот на 90 градусов

Задание 8. Операции с матрицами

Выполните операции с матрицами

A+B, A-B, A*B

% операции с точкой – векторные операции, выполняются поэлементно

A.*B % поэлементное умножение

A.*10 % умножить на число; заметим, что для умножения на скаляр не обязательно пользоваться векторной операцией, достаточно **A*10**

A.^2 %(каждый элемент матрицы, возведен в степень 2)

A^2 %(матрица, умноженная сама на себя (только для квадратных матриц!))

A=A+i

A' % транспонирование ли матрицы?

A.' % а так?

inv(B) – обратная матрица к B или равносильно:

B^(-1)

A=round(rand (3)) % округление до целого

sum(A), sum(A') % объясните, как работает функция sum.

sum(diag(A)), sum(diag(fliplr(A)))

sum(sum(A)), prod(A)

max(max(A)) % объясните, как работает функция

Задание 9. Справочная система (тематическая справка)

help elfun % Познакомьтесь со справкой по элементарным математическим функциям

help matfun % по изучению свойств матриц и специальных операций
help specfun % запомните, как вызывать справку по специальным функциям.

Операции сравнения и логические операции

Задание 10 Выполните следующие операции сравнения и объясните результат

123+456==579

3>9

[3 7 4]>[3 2 3] % поэлементное сравнение

t=ans

all(t) % если в t все логические единицы, то результат true (или 1)

any(t) % если в t есть хотя бы одна логические единицы, то результат true (или 1)

2<2.5000

7*8>=56

2^32<=65536

X=[1 2 3]

Y=['1','2','3']

Z=['1','2','A']

X==Y

X<Y

Z==Y

Как происходит сравнение?

Задание 11. Логические операции с массивами

Задайте массивы:

P=[1 1 1]

Q=[0 1 0]

R=[0 0 1]

Выполните следующие логические операции

Q|R

P&R

~Q

Посмотрите расширенную справку по логическим операциям, операциям отношения, операциям над множествами, для этого в командной строке наберите

help > (знак больше)

Задание 12. Операции над множествами

Придумайте самостоятельно примеры операций над множествами, используйте в качестве множеств векторы и следующие функции ML: unique, union, intersect, setxor, ismember.

Замечание. Язык нестрогой типизации обуславливает необходимость постоянного, насколько это требует логика программы, контроля типов.

Чтобы найти соответствующие функции контроля типов в закладке контекстного поиска в справке Help следует набрать *is**, как указано ниже

Enter index item: is*

В этом случае вы получите список всех функций контроля типов, например,
isequal – истина, если аргументы - матрицы равны поэлементно
isscalar – истина, если аргумент скаляр
isvector – истина, если аргумент вектор
Isrow – истина, если аргумент строка
Iscolumn - истина, если аргумент столбец
ismatrix – истина, если аргумент матрица

Операции с матрицами

Задание 13.

Задайте матрицы *r1* и *r2* одинакового размера

Объедините их двумя способами:

Rr=[r1;r2], Rc=[r1 r2] % аналогично выполните с помощью функции **cat**

Rr(:,end-1) = [] % удалена предпоследняя колонка

Rc(1,:)=1 % все элементы первой строки равны единице

Rr(1:2:end)=0 % каждый элемент матрицы с нечетным индексом обнуляется

spy(Rr) % графическое представление ненулевых элементов матрицы

5

Замечание. Матрица в памяти хранится как вектор, вытянутый по столбцам, поэтому обращение к элементам матрицы как к одномерному вектору корректно.

Задание 14. Элементарные математические функции

Задайте комплексные числа и найдите их характеристики:

d1=5+4i

d2=2-3j

abs(d1) % модуль числа

real(d1) % вещественную часть

imag(d1) % мнимую часть

conj(d2) % комплексно-сопряженное, сравните с **d1'**

Задайте вектор вещественных *x* и по результатам сформулируйте правила работы функций округления:

ceil(x), fix(x), floor(x), round(x)

проверьте работу функций

mod(x,y), rem(n,m)

сравните результаты. Уточните назначение этих функций с помощью справки.

Задание 15. Задание последовательностей (векторов-строк)

q=2:2:8 % начальное значение: шаг: конечное значение

r=3:3:12

v=q+r

f=v.^2, sum(f) % результат – скалярное произведение (v,v)

f=v^2 % устраните ошибку, представив второй способ вычисления скалярного произведения (v,v)

Элементы программирования. Скрипты, процедуры и функции.

*Скрипт файл (script) – внешний *.m файл, состоящий из последовательности ML-команд. Все переменные глобальные, в качестве первой команды следует использовать clear*

Задание 16. Скрипт-файл построения простейшего графика

Создайте script построения графика функции.

t=-pi:0.3:pi % изменение аргумента функции

e=sin(t) % массив значений функции

plot(t,e) % построение графика функции

Процедура без параметров используется в случае, когда целесообразна компиляция и требуется исключить интерактивность процесса выполнения скрипта. Процедура без параметров имеет в качестве шапки-заголовка процедуры следующую строку: function namef. Заметим, что при этом имя файла (namef.m) должно совпадать с именем namef, указанным в заголовке.

Задание 17. Процедура без параметров

Создайте скрипт как процедуру без параметров, параметры введите интерактивно по команде input (a=input('введите a') и т.п.) для вычисления площади треугольника по формуле Герона $s = \sqrt{p(p-a)(p-b)(p-c)}$, здесь p – полупериметр; a,b,c – длины сторон треугольника. Результат выведите на экран или сохраните в двоичном файле.

6

Задание 18. Процедура-функция

Создайте две процедуры-функции, которые вычисляют:

1. Максимальное значение элементов главной диагонали матрицы.
2. Сумму всех элементов матрицы, вне главной диагонали.

*Замечание. Первая строка функции начинается с заголовка. Заголовок функции должен иметь **один** выходной параметр (результат вычисления). Имя файла и имя функции также совпадают и начинаются с **буквы**. Синтаксис заголовка-функции следующий:*

function x=namefunction(parameter1, parameter 2)

% комментарий о назначении входных и выходных параметров

Задание 19. Процедура

Создайте процедуру, которая вычисляет корни квадратного уравнения x1, x2 уравнение задается своими коэффициентами a,b,c

*Заголовок процедуры аналогичен функции, но с **любым** количеством выходных параметров*

function [x,y]=namefunction(parameter1, parameter2)

% комментарий о назначении входных и выходных параметров