



ГУАП

guap.ru

Информатика. Информационные технологии

План лекции

- Язык программирования C++
 - Области видимости переменных C++
 - Ввод данных
 - Вывод данных
- Система компьютерной математики MatLab
 - Обзор системы
 - функции MatLab
 - m-файлы

Области видимости (scope)

- Область видимости – та часть программы в пределах которой возможен доступ к данной переменной
- большинство переменных является локальными – то есть доступ к ним возможен только в пределах одного блока, в котором они объявлены
- Глобальная переменная – переменная объявленная вне всех функций. Она доступна всюду в программе (global scope)
- Локальная переменная перекрывает глобальную

Область видимости переменной x

- Оператор разрешения видимости `::` - позволяет получить доступ к глобальной переменной
- Оператор `::` сообщает компилятору что нужно взять значение глобальной переменной `x`, а не локальной

```
1: #include <iostream.h>
2: #include <conio.h>
3: #pragma hdrstop
4: int x = 20;
5: void CountLoops(int);
6: int main(int, char**)
7: {
8:     int x = 40;
9:     int i = 0;
10:    cout << "In main program x = " << x << endl;
11:    bool done = false;
12:    while (!done) {
13:        int x;
14:        cout << endl << "Enter a number (-1 to exit): ";
15:        cin >> x;
16:        if (x != -1) {
17:            cout << endl << "In while loop x = " << x;
18:            CountLoops(++i);
19:        }
20:        else
21:            done = true;
22:    }
23:    cout << endl << "Global x = " << ::x << endl;
24:    cout << endl << "Press any key to continue...";
25:    getch();
26:    return 0;
27: }
28: void CountLoops(int x)
29: {
30:    cout << ", While loop has executed "
31:        << x << " times" << endl;
32: }
```

Global variable

Local variable

Local variable

Global scope

Local scope

Local scope

Многомодульные приложения

- Реальные приложения состоят из большого числа модулей и исходных файлов (source files)
- Глобальные переменные, объявленные в одном исходном файле, являются глобальными для этого файла, но невидимы в других модулях
- Для того, чтобы глобальные переменные были видны в других частях программы необходимо:
 - объявить переменную в одном из модулей
 - в других модулях объявить эту переменную с ключевым словом

extern int count;

Особенности использования глобальных переменных

- Глобальные переменные достаточно удобны, но в концепциях объектно-ориентированного программирования их использование не приветствуется
- Глобальные переменные постоянно занимают область памяти
- Использование глобальных переменных ухудшает безопасность написанной программы
- Рекомендуется использовать локальные переменные во всех возможных для этого ситуациях, стараясь свести к минимуму использование глобальных

Файловый вывод

- В языке C++ ввод-вывод осуществляется через объекты классов `ifstream` (для чтения данных) и `ofstream` (для вывода данных). Эти классы объявлены в заголовочном файле `fstream`.

```
#include <fstream>
using namespace std;
int main()
{
    ofstream fout;
    fout.open("file.txt");
    fout << "Hello, world!";
    fout.close();
    return 0;
}
```

Constant	Description
<code>ios::in</code>	открыть файл для чтения
<code>ios::out</code>	открыть файл для записи
<code>ios::ate</code>	при открытии переместить указатель в конец файла
<code>ios::app</code>	открыть файл для записи в конец файла
<code>ios::trunc</code>	удалить содержимое файла, если он существует
<code>ios::binary</code>	открытие файла в двоичном режиме

```
ofstream fout("file.txt", ios::app);
fout.open("file.txt", ios::app);
```


Наиболее часто возникающие ошибки в программах

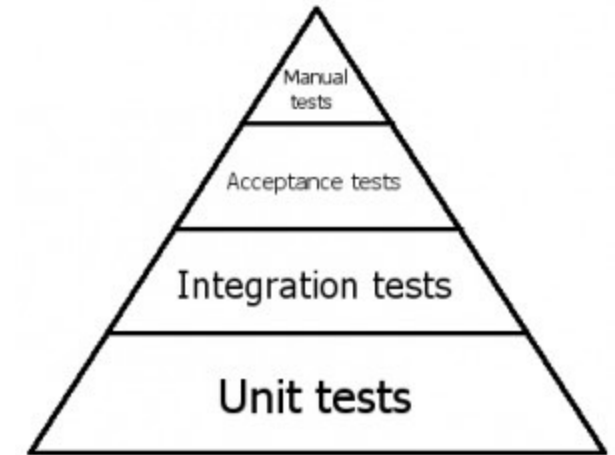
- 1. Ошибки исполнения (Runtime errors)
 - **данные ошибки возникают в процессе исполнения программ, часто они возникают только на стороне пользователя. Причиной ошибок может служить отсутствие установленных на компьютере программных компонент, несоответствие «железа» и.т.п.**
- 2. Логические ошибки (Logic errors)
 - Ошибки логического характера, вследствие которых программа может запуститься, но получаемый результат является некорректным. Они обычно возникают из – за:
 - некорректного использования логических операторов и переменных типа Boolean
 - создание бесконечных циклов
 - некорректное использование скобок
 - использование одинаковых имен переменных
- 3. Ошибки компиляции (Compilation errors)
 - Ошибка возникает в процессе построения машинного кода.
 - Данный тип ошибок не позволяет провести процедуру проверки и отладки программы, так как не удастся ее запустить.

Типичные ошибки программ

- 4. Синтаксические ошибки (Syntax errors)
 - **Нарушение грамматических правил построения программы**
 - **отсутствие скобок и точки с запятой**
- 5. Ошибки интерфейсов (Interface errors)
 - **ошибки взаимодействия сервер-клиент**
 - **ошибки передачи данных**
- 6. Ошибки ресурсов (Resource errors)
 - **недостаток постоянной или оперативной памяти**
 - **недостаток питания процессора,**
- 7. Ошибки вычисления (Arithmetic errors)
 - **ошибки вычислений и арифметические ошибки (деление на ноль, отсутствие численных решений)**

Тестирование, отладка (debugging)

- Отладка приложений - поиск и устранение ошибок в программе
 - 1 – Поиск синтаксических ошибок
 - 2 – Анализ кода
- Тестирование
 - **Модульное тестирование**, или **юнит-тестирование** (англ. unit testing) — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.
 - **Интегрированное тестирование** – тестирование объединенных модулей
 - **Acceptance test** – проверка на соответствие техническому заданию
 - **Ручное тестирование** – ручная проверка работоспособности

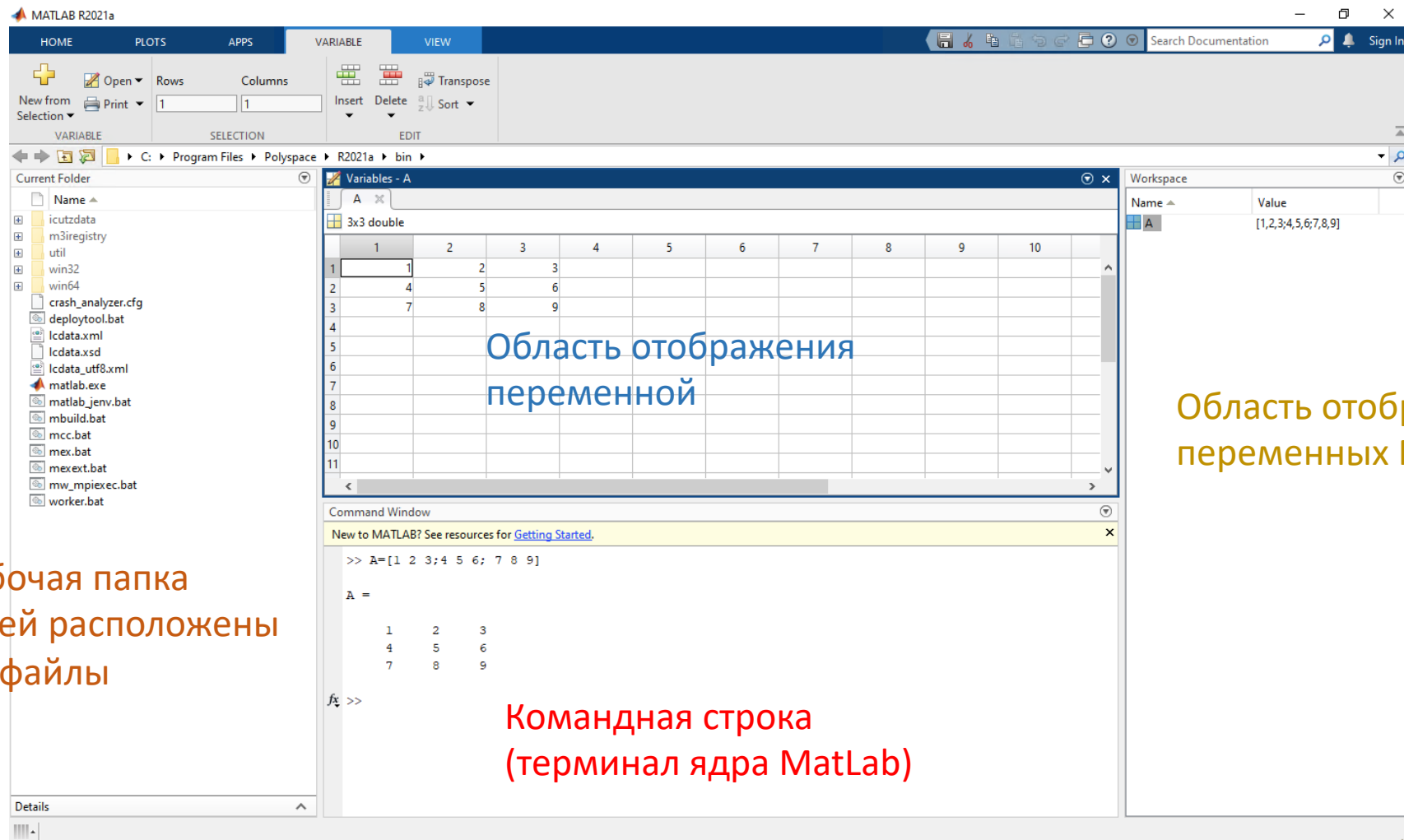


Пакет компьютерной математики MatLab



- MatLab – интерактивный, матрично ориентированный пакет прикладных программ, для выполнения научных и инженерных расчетов
- Значение аббревиатуры MatLab=Matrix+Laboratory (матричная лаборатория)
- MatLab – кроссплатформенный инструмент (работает на Windows, Linux, macOS)

Внешний вид графического интерфейса системы MatLab



Рабочая папка
В ней расположены
m- файлы

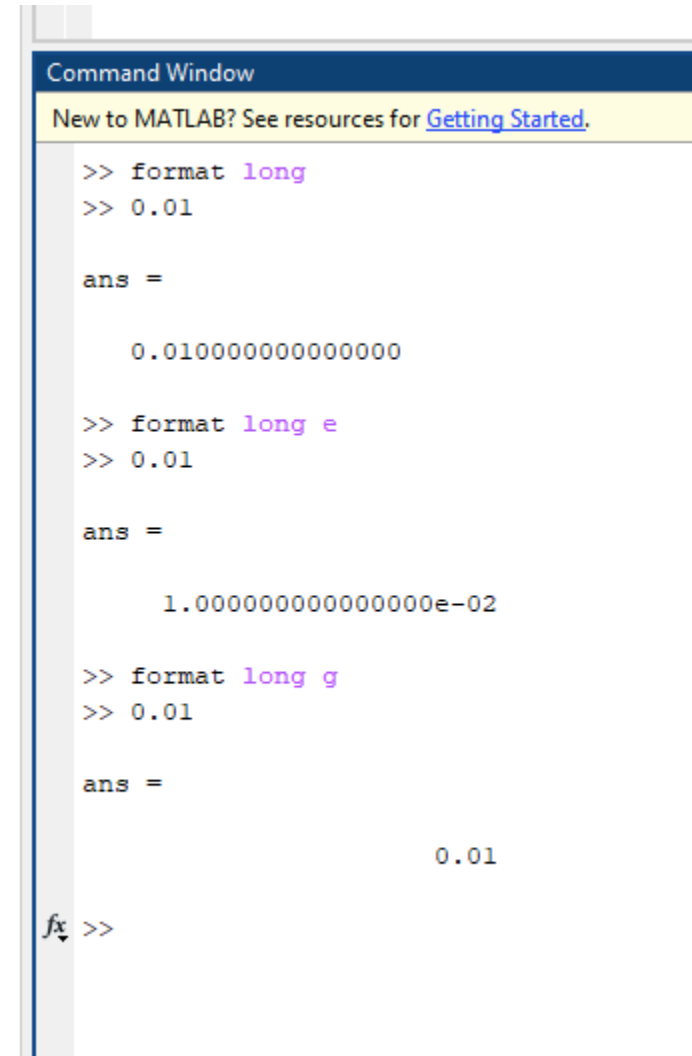
Область отображения
переменной

Область отображения
переменных MatLab

Командная строка
(терминал ядра MatLab)

Точность вычислений в MatLab

- Вычисления в системе MatLab производятся с двойной точностью
- Формат вывода данных может изменяться, в зависимости от указаний:
 - `format short` – фиксированная точка, 4 знака после точки (по умолчанию)
 - `format long` – фиксированная точка, 14 знака после точки
 - `format short e` – Научная нотация с 4 десятичными знаками
 - `format long e` – Научная нотация с 15 десятичными знаками
 - `format short g` – 3 знака после точки
 - `format long g` – 13 знаков после точки



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

>> format long
>> 0.01

ans =

    0.010000000000000000

>> format long e
>> 0.01

ans =

    1.0000000000000000e-02

>> format long g
>> 0.01

ans =

    0.01

fx >>
```

Логические операторы if, elseif, else

- Операторы сравнения

< - меньше

> - больше

<= - меньше или равно

>= - больше или равно

== - равно

~= - не равно

- Логические операторы

~ - не

& - и

| - или

```
Editor - Untitled*
for_while.m x Untitled* x +
1      %if statements
2      L=1:1:10;
3      choice = menu('Choose the case','case1','case2','case3','case4');
4      if choice==1
5          a = 1;
6          b=2;
7          c=3;
8      elseif choice==2
9          a = 6;
10         b=8;
11         c=9;
12     else
13         a = 2;
14         b=6;
15         c=8;
16     end
17     x=(a/b)^c.*L;
18     plot(L,x)
19     %switch case
20     method = 'Bilinear';
21     switch lower(method)
22         case {'linear','bilinear'}
23             disp('Method is linear')
24         case 'cubic'
25             disp('Method is cubic')
26         case 'nearest'
27             disp('Method is nearest')
28         otherwise
29             disp('Unknown method.')
30     end
```

Циклы в MatLab

for i=max:iterator:min

< statement>

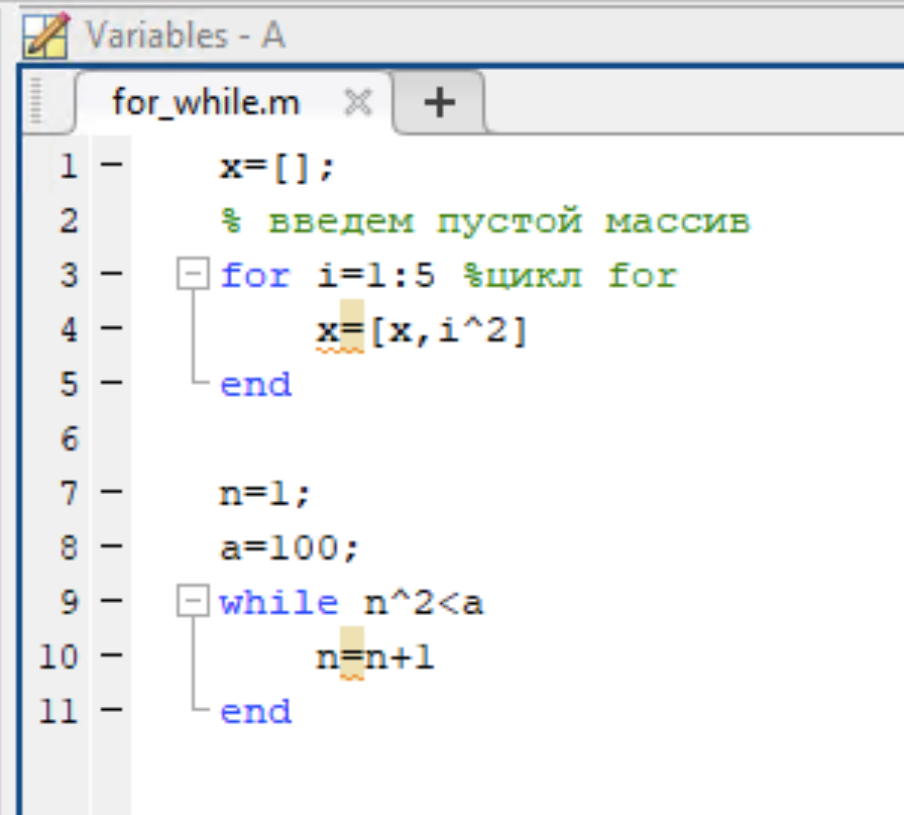
end

while <condition>

< statement>

end

x=[]; for i=5:-1:1,x=[x,i^2], end



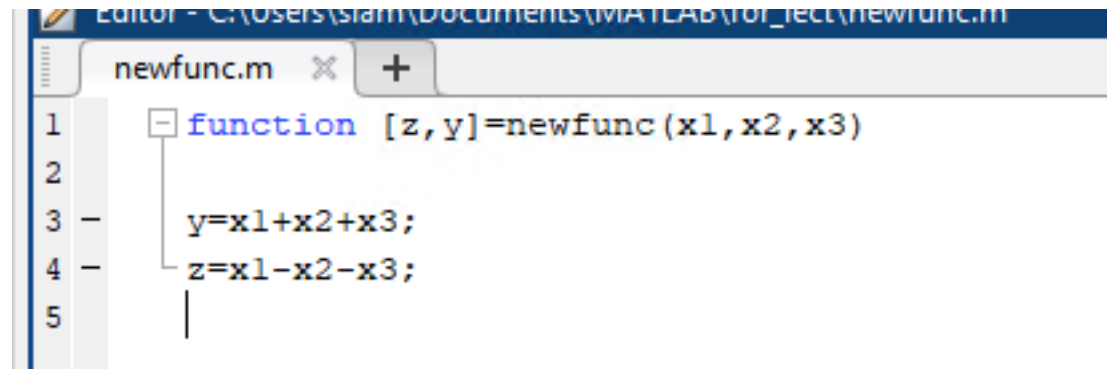
```
Variables - A
for_while.m x +
1 - x=[];
2 - % введем пустой массив
3 - for i=1:5 %цикл for
4 -     x=[x,i^2]
5 - end
6
7 - n=1;
8 - a=100;
9 - while n^2<a
10 -     n=n+1
11 - end
```

Функции в MatLab

- Система MatLab содержит большое количество встроенных функций написанных разработчиком, обращение к которым производится следующим образом: *function(argument)*
- Функции MatLab могут возвращать несколько значений
- Скалярные функции: `sin()`, `asin()`, `exp()`, `abs()`, `round()`, `cos()`, `acos()`, `log()`, `tan()`, `atan()`
- Векторные функции (аргументы функции это вектора), в случае если данные функции действуют на матрицу, они действуют на каждый столбец: `max()`, `min()`, `sum()`, `sort()`
- Матричные функции: `det()`, `rank()`

m - файлы

- MatLab – может выполнять последовательность операторов, записанных в файл. Формат таких файлов для <file>.m
- Переменные в программе являются глобальными
- Внутри программы можно ссылаться на другие m файлы
- Файлы – функции. Назвав m- файл именем функции, описанной в нем к этой функции можно обращаться как к встроенным функциям MatLab



```
Editor - C:\Users\slam\Documents\MATLAB\101_tect\newfunc.m
newfunc.m x +
1 function [z,y]=newfunc(x1,x2,x3)
2
3     y=x1+x2+x3;
4     z=x1-x2-x3;
5
```