

## 2.2 Многооконный редактор–отладчик с окнами редактирования m-файлов

В среде программирования MATLAB особое значение имеют файлы двух типов - с расширениями .mat и .m [14]. Первые являются бинарными файлами, в которых могут храниться значения переменных, вторые представляют собой текстовые файлы, содержащие внешние программы, определения команд и функций MATLAB.

Именно к ним относится большая часть команд и функций, в том числе задаваемых пользователем для решения своих специфических задач. Многооконный редактор–отладчик с пустым окном редактирования m-файлов можно вызвать командой Edit из командной строки или командой меню File → New → M-file (рисунок 2.1).

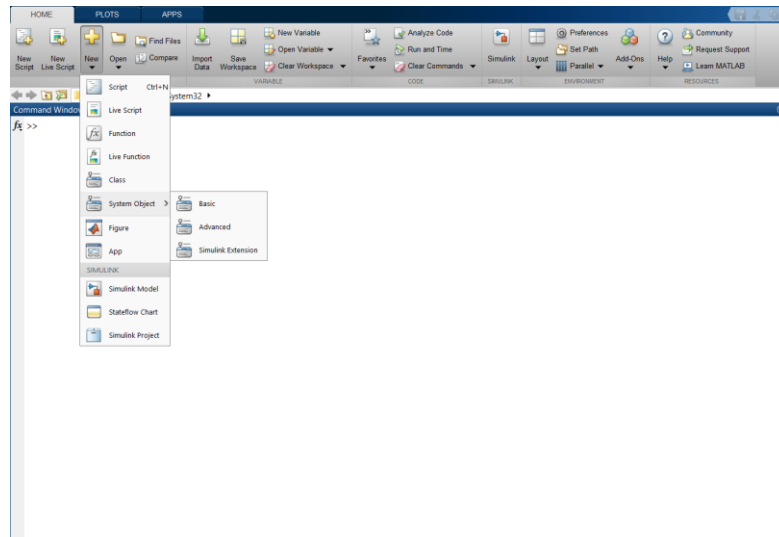


Рисунок 2.1. Многооконный редактор–отладчик

После этого в окне редактора можно создать свой файл, а также пользоваться средствами его отладки и запуска. Для запуска файла его необходимо записать на диск, используя команду Save as в меню File редактора. Редактор отладчик m-файлов выполняет синтаксическую проверку программного кода по мере ввода текста. При этом используется следующее цветовое выделение:

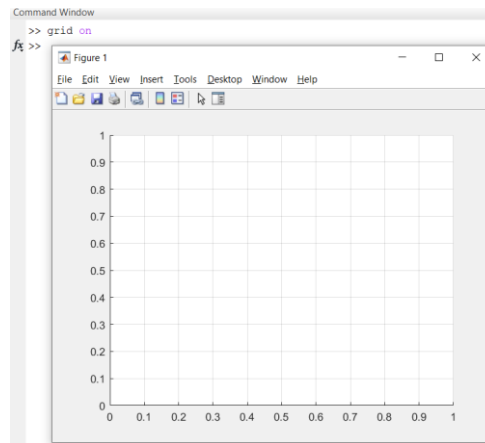
- ключевые слова языка программирования - синий цвет;
- операторы, константы и переменные - черный цвет;
- комментарии после знака % - зеленый цвет;
- символьные переменные (в апострофах) - коричневый цвет;
- синтаксические ошибки - красный цвет.

Благодаря цветовому выделению вероятность синтаксических ошибок резко снижается m-файлы, создаваемые редактором-отладчиком, делятся на два класса: файлы-сценарии, не имеющие входных параметров и файлы-функции, имеющие входные параметры. Файл-сценарий, именуемый также script-файлом, является просто записью серии команд без входных и выходных параметров. Он имеет следующую структуру:

%Основной комментарий  
%Дополнительный комментарий  
Тело файла с любыми выражениями.

Важны следующие свойства файлов – сценариев:

- 1) они не имеют входных и выходных аргументов;
- 2) работают с данными из рабочей области;
- 3) в процессе выполнения не компилируются;
- 4) представляют собой зафиксированную в виде файла последовательность операций, полностью аналогичную той, что используется в сессии. В качестве примера рассмотрим файл-сценарий, приведенный на рисунке 2.2.



**Рисунок 2.2. Создание файла-сценария в MATLAB**

Первые три строки здесь - это комментарий, остальные - тело файла. Обратите внимание на возможность задания комментария на русском языке. Знак % в комментариях должен начинаться с первой позиции строки. Необходимо отметить, что такой файл нельзя запустить без предварительной подготовки, сводящейся к заданию значений переменным `xmin` и `xmax`, использованным в теле файла. Это следствие первого свойства файлов-сценариев - они работают с данными из рабочей области. Имена файлов-сценариев нельзя использовать в качестве параметров функций, поскольку файлы-сценарии не возвращают значений. Можно сказать, что файл-сценарий - это простейшая программа на языке программирования MATLAB.

М-файл-функция является типичным объектом языка программирования системы MATLAB [14]. Одновременно он является полноценным модулем с точки зрения структурного программирования, поскольку содержит входные и выходные параметры и использует аппарат локальных переменных. Структура такого модуля с одним выходным параметром выглядит следующим образом:

```
function var = f_name(Список_параметров)
%Основной комментарий
%Дополнительный комментарий
Тело файла с любыми выражениями
var=выражение
```

### **2.3 Свойства М-файл-функции**

М-файл-функция имеет следующие свойства [4-5]:

- 1) он начинается с объявления `function`, после которого указывается имя переменной `var` - выходного параметра, имя самой функции `f_name` и список ее входных параметров;
- 2) функция возвращает свое значение и может использоваться в математических выражениях;
- 3) все переменные, имеющиеся в теле файла-функции, являются локальными, т. е. действуют только в пределах тела функции;
- 4) файл-функция является самостоятельным программным модулем, который общается с другими модулями через свои входные и выходные параметры;
- 5) правила вывода комментариев те же, что у файлов-сценариев;
- 6) при обнаружении файла-функции он компилируется и затем исполняется, а созданные машинные коды хранятся в рабочей области программы MATLAB.

Последняя конструкция `var = выражение` вводится, если требуется, чтобы функция возвращала результат вычислений. Приведенная форма файла-функции характерна для функции с одним выходным параметром. Если выходных параметров больше, то они указываются в квадратных скобках после слова `function`. При этом структура модуля имеет следующий вид:

```
function [var1,var2,...] = f_name(Список_параметров)
%Основной комментарий
%Дополнительный комментарий
Тело файла с любыми выражениями
var1=выражение
var2=выражение
```

Если функция используется как имеющая единственный выходной параметр, но имеет ряд выходных параметров, то для возврата значения будет использоваться первый из них. Это зачастую

ведет к ошибкам в математических вычислениях. Поэтому, как отмечалось, данная функция используется как отдельный элемент программ вида [var1, var2] = f\_name (список\_параметров). После его применения переменные выхода var1, var2 становятся определенными и их можно использовать в последующих математических выражениях и иных сегментах программы. Для организации диалогового ввода и вывода используются следующие операторы, представленные в таблице 2.2.

Приведем простой пример диалоговой программы, которая служит для многократного вычисления длины окружности по вводимому пользователем значению радиуса r (рисунок 2.3).

Для организации ветвлений служат условные операторы. Конструкции условных операторов: 1) if <условие> <операторы> end.

Операторы (тело выражения) выполняются только в том случае, если условие истинно, если условие ложно, то тело выражения не выполняется. 2) if <условие> <операторы 1> else <операторы 2> end.

Таблица 2.2. Операторы диалогового ввода/вывода

Оператор	Синтаксис	Назначение
INPUT	x = input('<приглашение>')	Для ввода данных с клавиатуры
DISP	disp (<переменная или текст в апострофах>)	Для вывода на экран монитора

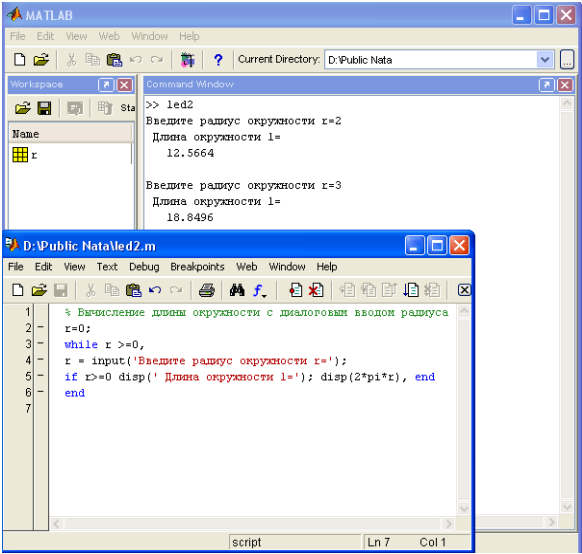


Рисунок 2.3. Пример диалоговой программы

Если ход программы должен изменяться в зависимости от нескольких условий, то следует использовать полную конструкцию if-elseif-else. Каждая из ветвей elseif в этом случае должна содержать условие выполнения блока операторов, размещенных после нее. Важно понимать, что условия проверяются подряд, первое выполненное условие приводит к работе соответствующего блока, выходу из конструкции if-elseif-else и переходу к оператору, следующему за end. У последней ветви else не должно быть никакого условия.

Операторы, находящиеся между else и end, работают в том случае, если все условия оказались невыполненными.

Например, составить файл-функцию для вычисления кусочно-заданной функции:

$$f(x) = \begin{cases} 1 - e^{-1-x}, & x < -1; \\ x^2 - x - 2, & -1 \leq x \leq 2; \\ 2 - x, & x > 2. \end{cases}$$

В программе MATLAB могут применяться следующие операторы сравнения, приведенные в таблице 2.3.

Таблица 2.3. Операторы сравнения

Символ	Назначение	Имя функции
<	Меньше	Lt
>=	Больше или равно	Ge
>	Больше	Gt
<=	Меньше или равно	Le
=	Равно	Eq
~=	Не равно	Ne

Операции (==, ~=) проводят сравнение вещественных и мнимых частей комплексных чисел, а операции (>, <, >=, <=) – только вещественных частей. Логические операции можно записывать в виде функций (таблица 2.4).

Таблица 2.4. Логические операции

Символ	Назначение	Имя функции
&	Логическое «и»	And
	Логическое «или»	Or
~	Отрицание	Not

Результатом логических операций являются числа 0 (false) и 1(true). В MATLAB есть две разновидности операторов цикла – условный и арифметический [14].

Для повторения операторов нефиксированное число раз используется оператор цикла с предусловием:

```
while <условие>
<операторы>
end.
```

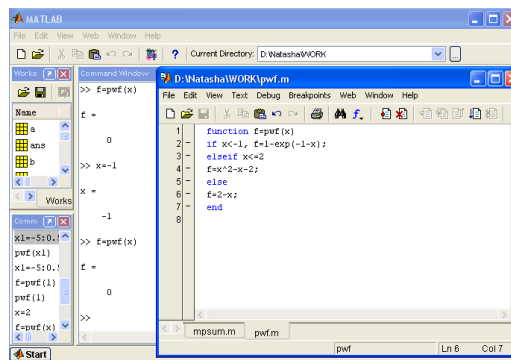


Рисунок 2.4. Листинг программы для вычисления значения функции

Операторы выполняются, если переменная <условие> имеет ненулевые элементы. Арифметический оператор цикла имеет следующий вид:

```
for <имя> = <НЗ>: <Шаг>: <КЗ>
<операторы>
end,
```

где <имя> – имя управляющей переменной цикла, <НЗ> – начальное значение управляющей переменной, <КЗ> – конечное значение управляющей переменной, <Шаг> – приращение значений переменной <имя> в ходе ее изменения от значения <НЗ> до значения <КЗ>. Если параметр <Шаг> не указан, по умолчанию его значение принимается равным единице.

При работе с циклом for допустимо использование оператора прерывания цикла break. При работе данного оператора работа цикла завершается, и управление передается на следующий после конца цикла оператор.

Ход работы программы может определяться значением некоторой переменной (переключателя). Такой альтернативный способ ветвления программы основан на использовании оператора переключения switch. Оператор switch содержит блоки, начинающиеся со слова case, после каждого case записывается через пробел то значение переключателя, при котором выполняется данный блок. Последний блок начинается со слова otherwise, его операторы работают в том случае, когда ни один из блоков case не был выполнен. Если хотя бы один из блоков case выполнен, то происходит выход из оператора switch и переход к оператору, следующему за end.

Предположим, что требуется найти количество единиц и минус единиц в заданном массиве и, кроме того, найти сумму всех элементов, отличных от единицы и минус единицы. Листинг программы содержит файл-функцию, которая по заданному массиву возвращает число минус единиц в первом выходном аргументе, число единиц - во втором, а сумму - в третьем (рисунок 2.5).

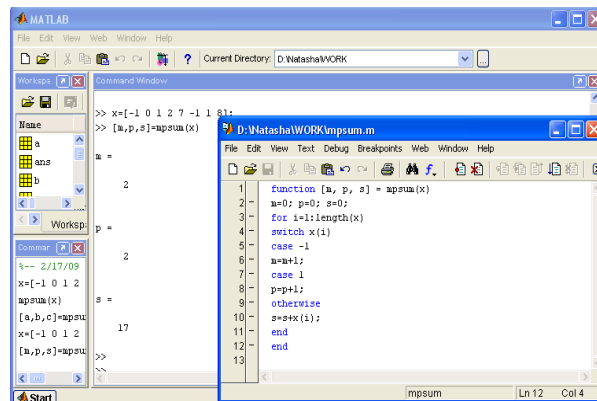


Рисунок 2.5. Листинг программы

Для остановки программы используется оператор `pause`.

Он используется в следующих формах:

- a) `pause` - останавливает вычисления до нажатия клавиши;
- b) `pause(N)` - останавливает вычисления на `N` секунд;
- c) `pause on` - включает режим отработки пауз;
- d) `pause off` - выключает режим отработки пауз.