

Тема 4.3

Программирование на языке MATLAB

Вопросы для изучения

- 4.10 Программирование алгоритмов линейной структуры
- 4.11 Организация диалога с пользователем.
- 4.12 Ввод-вывод данных по заданному формату
- 4.13 Функции обработки ошибок

4.10 Программирование алгоритмов линейной структуры

Линейный алгоритм – набор команд, выполняемых последовательно во времени друг за другом.

При составлении линейных программ с помощью М-файлов в Matlab одним из основных операторов является оператор присваивания.

Общий вид оператора присваивания:

Имя_переменной = Выражение

где **Имя_переменной** - имя простой переменной, структурированной переменной (вектора, матрицы), имя функции;

Выражение - арифметическое, логическое или строковое выражение.

Тип переменной определяется системой автоматически по типу выражения, поэтому нет необходимости при программировании следить за соответствием типов данных в операторе присваивания. Если выражение содержит и арифметические и строковые элементы, то переменная будет численной, т.е. предпочтение отдается числовому типу данных.

Пример

$$A = \cos(x) + c - d^2 * p^2 + 4.92$$

N = 'номер формулы'

$$R = (x > 5) \& (x \leq 10)$$

Соответственно использования арифметического выражения, во втором – символьного, в третьем – логического.

4.11 Организация диалога с пользователем

При программировании часто возникает необходимость вводить исходные данные не с помощью оператора присваивания, а с помощью операторов ввода в диалоговом режиме. Преимущества такого ввода очевидны: при вводе новых исходных данных нет необходимости вносить изменения в программу.

В MatLab в качестве оператора ввода используется функция **input**, которую, в силу ее значимости при программировании, принято называть **оператором ввода**.

Она имеет следующий общий вид:

$$\text{ИМЯ} = \text{input}(\text{Символьная константа})$$

где **ИМЯ** – это имя простой переменной,

Символьная константа – любой набор символов, заключенный в апострофы. Символьная константа, как правило, разъясняет смысловое назначение вводимой переменной.

Например:

$$\begin{aligned} S &= \text{input}(\text{'Задайте площадь'}) \\ A &= \text{input}(\text{'Задайте значение A'}) \end{aligned}$$

Оператор выполняется следующим образом:

- в командном окне выводится набор символов, стоящую в скобках после input (символьная константа);
- выполнение программы приостанавливается и компьютер переходит в режим ожидания;
- пользователь вводит константу;
- введенная константа помещается в оперативной памяти в переменную, стоящую слева в операторе input.

При запуске на выполнение программы, содержащей оператор ввода следует учитывать, что пока пользователь не ввел константу в ответ на запрос своей программы, оператор ввода продолжает свою работу. Система Matlab в это время блокирует выход и закрытие окна рабочего стола.

Если необходимо вывести данные на экран дисплея в определенной последовательности, отличной от последовательности их вычисления, применяется функция **disp**, которую принято называть **оператором вывода**.

Оператор имеет следующий общий вид:

`disp(Выражение)`

где **Выражение** – это арифметическое, логическое или символьное выражение, частным случаем которого являются константы или переменные любого типа.

Например

`disp('Результаты')`

`disp(summa)`

`disp(5+6)`

`disp(a)`

`disp((x>5)&(x<=10))`

Следует помнить, что Matlab выводит в командном окне значение переменной, стоящей слева в операторе присваивания, если оператор не заканчивается символом «;», поэтому если в программе используется оператор вывода, нужно подавлять дублирование вывода, указывая символ «;» в конце оператора присваивания.

Каждый новый оператор `disp` выполняет вывод с новой строки командного окна, например:

Фрагмент программы

```
a=10;  
b=5;  
k=2;  
d=1;  
c=a-b+k*d;  
disp('результат=')  
disp(c);
```

Командное окно

```
результат=  
7
```

Если необходимо вывести несколько данных в одной строке, нужно сформировать из них вектор-строку, который будет использоваться за-тем в выражении в операторе `disp`, например:

Фрагмент программы

```
a=10;  
b=5;  
k=2;  
d=1;  
c=a-b+k*d;  
r='результат='  
r1=' при a=';  
x=[r, c, r1, a];  
disp(x);
```

Командное окно

```
результат=7 при a=10
```



Пример реализации линейного алгоритма в командном и программном режиме

4.12 Ввод-вывод данных по заданному формату

Система MATLAB представляет на машинном уровне все действительные числа заданные мантиссой и показателем степени, например, $2.85093E+11$, где буквой E обозначается основание степени равное 10.

Этот основной тип данных носит название `double`. MATLAB по умолчанию использует формат `short` для вывода вещественных чисел, при котором показываются только четыре десятичных цифры после запятой.

Если требуется полное представление вещественного числа, введите с клавиатуры команду:

```
>> format long
```

далее наберите имя переменной и нажмите клавишу Enter теперь все результаты вычислений будут показываться с такой высокой точностью в течение данного сеанса работы в среде системы MATLAB.

Если требуется до прекращения текущего сеанса работы вернуться к формату `short` в командном окне, нужно исполнить команду:

```
>> format short
```

Целые числа показываются системой в командном окне в виде целых чисел.

4.13 Функции обработки ошибок

В тех случаях, когда возникают различного рода ошибки, желательно, чтобы система реагировала на них специальным образом.

Возможности обработки ошибок в системе MATLAB позволяют прикладной программе проверить специфические условия возникновения ошибки и выдать код ошибки в зависимости от ситуации.

Вывод на экран ошибок и предупреждающих сообщений.

Для вывода на экран терминала информации об ошибке предназначены функции `error` и `warning`.

Функция вывода сообщения об ошибке имеет следующий синтаксис:

```
error('<сообщение об ошибке>')
```

Если эта функция ошибки вызывается внутри М-файла, то ошибка отображается в текущей строке, а выполнение М-файла прекращается.

Пример

Допустим, что в текст М-файла `my.m` включено условие

```
if a < 1
    error('n должно быть больше или равно 1.')
end
```

Для `n`, равного 0, следующий текст появляется на экране, а выполнение М-файла прекратится:

n должно быть больше или равно 1.

Предупреждения системы MATLAB аналогичны сообщениям об ошибках, за исключением того, что выполнение программы не прекращается. Для вывода на экран предупреждающих сообщений предназначена функция `warning`, имеющая следующий синтаксис:

```
warning('<строка_предупреждения>').
```

Пример

```
n = 0;
```

```
if n < 1
```

```
    warning('n должно быть больше или равно 1.')
```

```
end
```

Warning: n должно быть больше или равно 1.