

**Тема 4.5**

**Программирование  
на языке MATLAB**

## **Вопросы для изучения**

4.17 Программирование циклических алгоритмов

4.18 Операторы с известным числом повторений и с условием

4.19 Операторы управления работой цикла

4.20 Вложенные циклы

## 4.17 Программирование циклических алгоритмов

**Циклический алгоритм** – алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов. Многократно повторяющиеся участки называются циклами или телом цикла цикла.

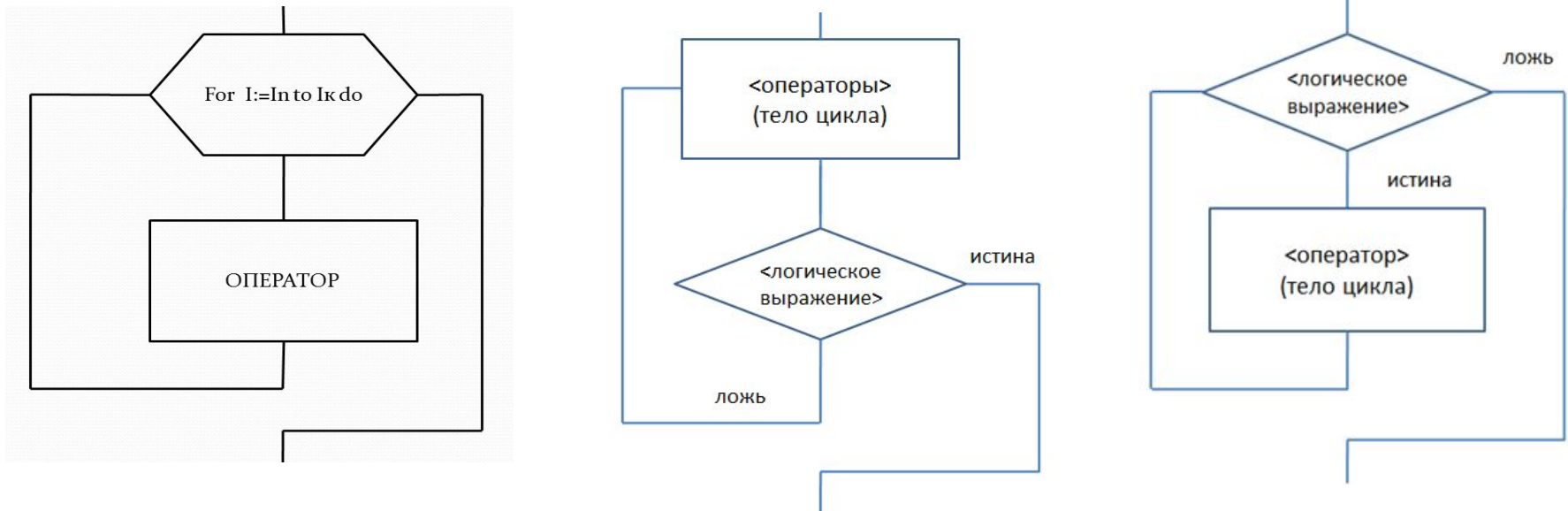


Рисунок 4.171- Циклические алгоритмы

Операторы цикла предназначены для программирования циклических алгоритмов, они изменяют естественный ход выполнения программы и относятся к операторам управления.

Цикл может управляться явно выраженной переменной цикла, имеющей параметры, либо цикл не содержит переменной цикла и управляется условием.

Операторы цикла можно разделить по тому же принципу:

- для программирования циклов с переменной цикла и параметрами;
- для программирования циклов без явно выраженной переменной цикла.

Операторы цикла в Matlab можно классифицировать следующим образом:

- оператор цикла с параметрами `for`
- оператор цикла с условием `while`

Операторы являются взаимозаменяемыми, выбор для применения того или иного оператора зависит от программиста.

## 4.18 Операторы с известным числом повторений и с условием

Оператор цикла **for** предназначен для программирования циклических алгоритмов, когда переменная цикла явно выражена и изменяется от начального значения до конечного значения с постоянным шагом. В языке Matlab цикл **for** предназначен для выполнения заданного числа повторяющихся действий:

Форма 1.

**for** Имя ПЦ= Начальное значение ПЦ : Шаг изменения ПЦ : Конечное значение ПЦ

Инструкции

**End**

Форма 2.

**for** Имя ПЦ = Начальное значение ПЦ : Конечное значение ПЦ

Инструкции

**End**

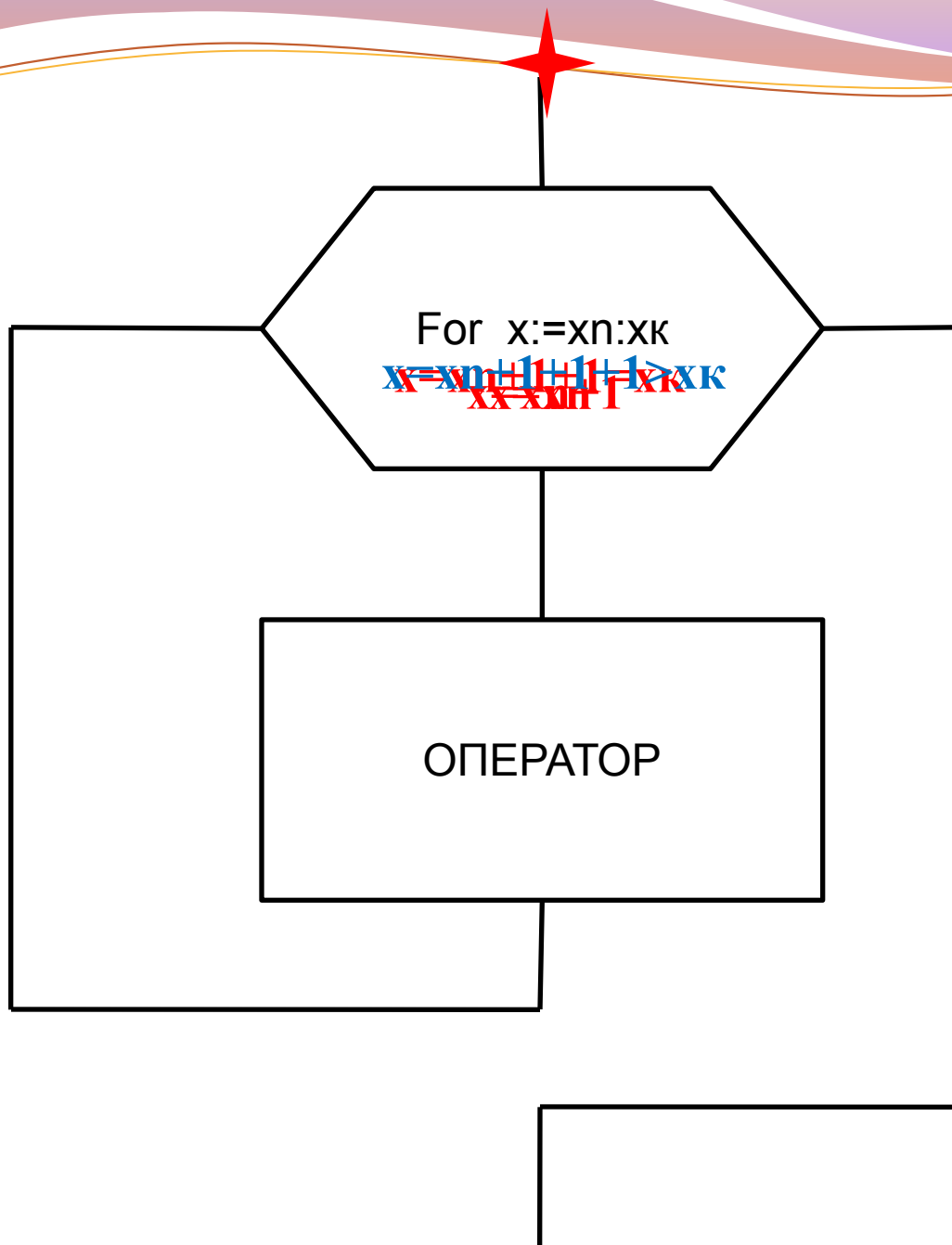
где Имя ПЦ– идентификатор переменной цикла;  
Начальное значение ПЦ– ее начальное значение переменной цикла;  
Конечное значение ПЦ– конечное значение переменной цикла,  
Шаг изменения ПЦ – шаг, на который увеличивается значение переменной цикла при повторении цикла;  
Инструкции – оператор или группа операторов составляющих тело цикла.

Во 2 форме оператора for шаг изменения переменной цикла равен 1.

Переменная цикла может принимать не только целые, но и вещественные значения любого знака.

Работа:

- 1) проверяется условие Начальное значение ПЦ  $\leq$  Конечное значение ПЦ,
- 2) если условие не выполняется, то оператор цикла прекращает свою работу, и выполняется следующий за ним оператор программы;
- 3) если условие выполняется, то переменной цикла присваивается ее начальное значение (или принимается следующее с заданным шагом);
- 4) выполняется тело цикла (инструкции).
- 5) затем переменная цикла увеличивается на шаг и выполняется этап 4
- 6) шаг 5 повторяется пока значение ПЦ  $\leq$  Конечное значение ПЦ в противном случае оператор цикла прекращает свою работу, и выполняется следующий за ним оператор программы.



Порядок выполнения оператора цикла(2 форма) тот же, только переменная цикла увеличивается на шаг, равный 1.

Параметры цикла могут быть константами, переменными или выражениями. Если параметры цикла заданы выражениями, все переменные, входящие в состав выражений должны быть определены заранее. Если параметры цикла заданы некорректно, то оператор цикла не выполняется ни разу.

Корректное задание параметров цикла:

Начальное значение ПЦ  $\leq$  Конечное значение ПЦ при Шаг изменения ПЦ  $>0$ ,  
Начальное значение ПЦ  $\geq$  Конечное значение ПЦ при Шаг изменения ПЦ  $<0$ .

Количество повторений цикла вычисляется по формуле:

$(\text{Конечное значение ПЦ} - \text{Начальное значение ПЦ}) / \text{Шаг изменения ПЦ} + 1$ .



```
for k=1 : 5
    disp(k)
end
```

```
for s=10 :-1: 6
    disp(s)
end
```

```
c=8; d=11;
for i=c +d : 50
    a=i+5
    b=a+i
end
```

Пример: найти сумму четных чисел от 1 до 100

Оператор **while** предназначен для программирования любых циклов, где проверка условия повторения цикла выполняется перед выполнением рабочей части цикла и действий, число которых заранее не известно, но известно условие продолжения цикла:

**while** условие

Инструкции

**end**

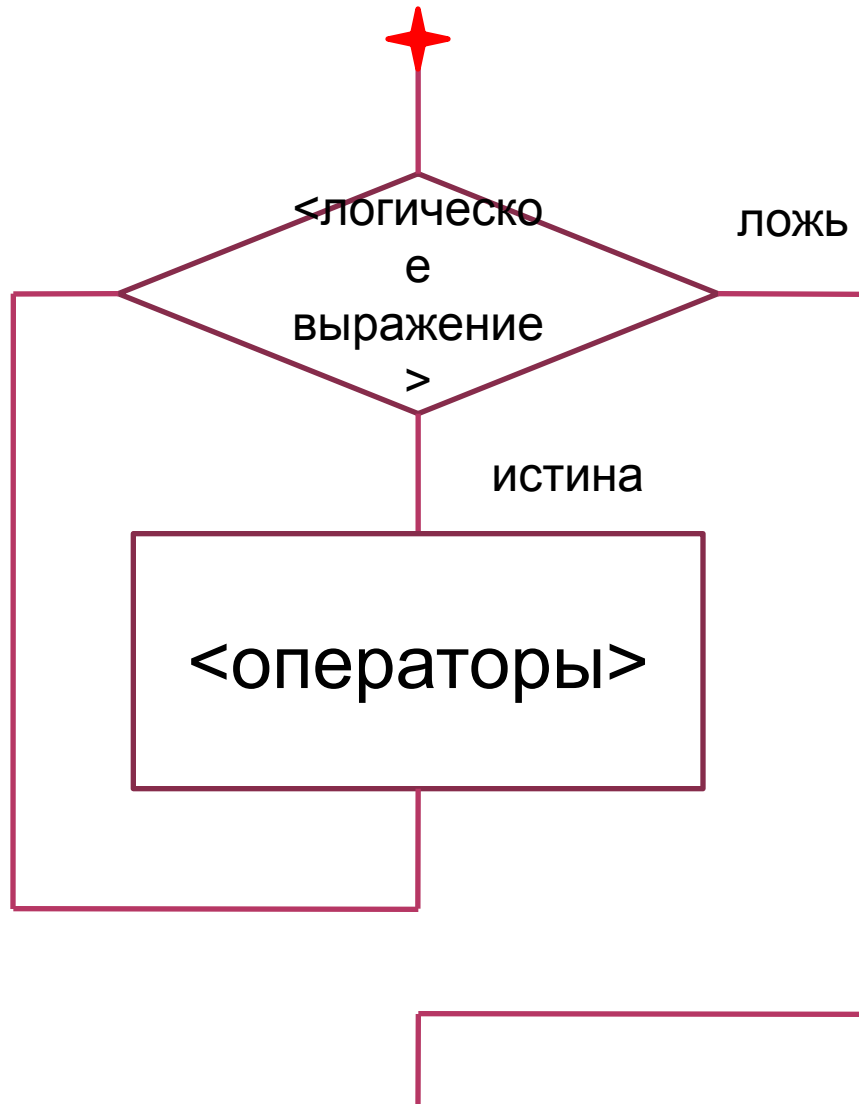
где условие – логическое выражение, значение или константа

Инструкции–оператор или группа операторов рабочей части (тела) цикла.

Работа:

- 1) проверяется (вычисляется) значение логического выражения;
- 2) пока оно истинно «True» (логическая единица), выполняется тело цикла (инструкции);
- 3) повторяются шаги 1 и 2;
- 4) если логическое выражение принимает ложное значение «False» (логический ноль), то работа оператора прекращается и выполняется следующий за оператором **while** оператор программы.

# Изображение оператора while в блок-схемах



## Рекомендации по использованию оператора while.

Если переменная цикла явно выражена, то:

- нужно присвоить ей начальное значение до оператора цикла;
- необходимо предусмотреть ее изменение в теле цикла;
- она должна использоваться в логическом выражении условия повторения цикла.

Если логическое выражение в заголовке цикла while заведомо ложно, то цикл может не выполниться ни разу.

Пример: найти сумму чисел от 1 до 100

```
S =0;  
i=1;  
while i <= 20  
S=S+i;  
i=i+1;  
end  
disp(S);
```

Проверить требования о недопустимости входа в цикл минуя заголовок и на изменение переменной цикла for внутри цикла

## 4.19 Операторы управления работой цикла

В управляющих структурах, в частности в циклах `for` и `while`, часто используются операторы, влияющие на их выполнение.

Так, оператор **`break`** может использоваться для досрочного прерывания выполнения цикла. Как только он встречается в программе, цикл прерывается.

Оператор **`continue`** передает управление в следующую итерацию цикла, пропуская операторы, которые записаны за ним, причем во вложенном цикле он передает управление на следующую итерацию основного цикла.

```
S = 0;
i = 1;
while i <= 20
    S = S + i;
    i = i + 1;
    if S >= 10
        break;
    end
end
disp(S);
```

## 4.20 Вложенность циклов

В теле любого оператора цикла могут находиться другие операторы цикла. При этом цикл, содержащий в себе другой, называют внешним, а цикл, находящийся в теле первого — внутренним (вложенным).

Таких вложенных циклов во внешнем цикле может быть несколько.

Правила организации внешнего и внутреннего циклов такие же, как и для простого цикла.

При программировании вложенных циклов необходимо соблюдать следующее дополнительное условие: все операторы внутреннего цикла должны полностью располагаться в теле внешнего цикла.

При вложении циклов `for` внутренний цикл выполняется полностью от начального до конечного значения параметра, при неизменном значении параметра внешнего цикла. Затем значение параметра внешнего цикла изменяется на единицу, и опять от начала и до конца выполняется вложенный цикл. И так до тех пор, пока значение параметра внешнего цикла не станет больше конечного значения, определенного в операторе `for` внешнего цикла.

Аналогично выполняется работа итерационных циклов (while) за исключением того, что проверка условия выполнения внешнего цикла выполняется только после всех итераций для выполнения условия внутреннего цикла.

Возможна комбинация разных операторов цикла при использовании вложенности условия их работы определяются комбинированием вышеуказанных правил.

Пример ввода двумерного массива с помощью операции индексации

```
n=input('ввести размерность массива n= ');
```

```
for i=1:n
```

```
    for j=1:n
```

```
        d(i,j)=i*j;
```

```
    end
```

```
end
```

```
disp(['получена матрица размером' num2str(n) 'x' num2str(n)]);
```

```
disp(d);
```





Рассмотреть комбинации вложенности разных операторов цикла? Привести блок-схемы алгоритмов.