

# Виды алгоритмов

1. **Линейные алгоритмы**
2. **Разветвляющие алгоритмы**
3. **Циклические алгоритмы**
4. **Построение блок - схем**

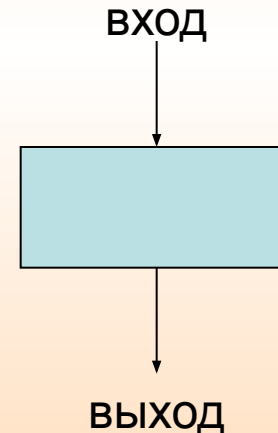
# 1. Линейные алгоритмы

**Линейный алгоритм** – это алгоритм, в котором блоки выполняются последовательно сверху вниз от начала до конца.

Линейный алгоритм состоит из последовательности операций, выполняющихся только один раз в порядке следования.

На практике линейные алгоритмы в чистом виде встречаются редко: при расчете арифметических и алгебраических выражений, при расчете по формулам, при решении ряда бытовых задач

**Базовая структура линейного алгоритма:**



Рассмотрим составление схем линейных алгоритмов на конкретных примерах:

# Задача

Составить блок схему алгоритма вычисления периметра  $P$  и площади  $S$  квадрата со стороной длины  $A$ .

## Словесно – формульное описание:

1. Ввод  $A$ , перейти к п.2
2. Вычисление  $P:=4 \cdot A$ ,  
перейти к п.3
3. Вычисление  $S:=A \cdot A$ ,  
перейти к п.4
4. Вывод  $A, P, S$
5. Вычисления прекратить.

## Псевдокод

*алг «Вычисление периметра и площади квадрата»*

*нач*

*запрос («сторона квадрата =»,  $A$ )*

*$P:=4 \cdot A$*

*$S:=A \cdot A$*

*вывод («сторона квадрата =»,  $A$ )*

*вывод («периметр квадрата =»,  $P$ )*

*вывод (« площадь квадрата =»,  $S$ )*

*кон*

## 1. Математическое обоснование

A – известная переменная

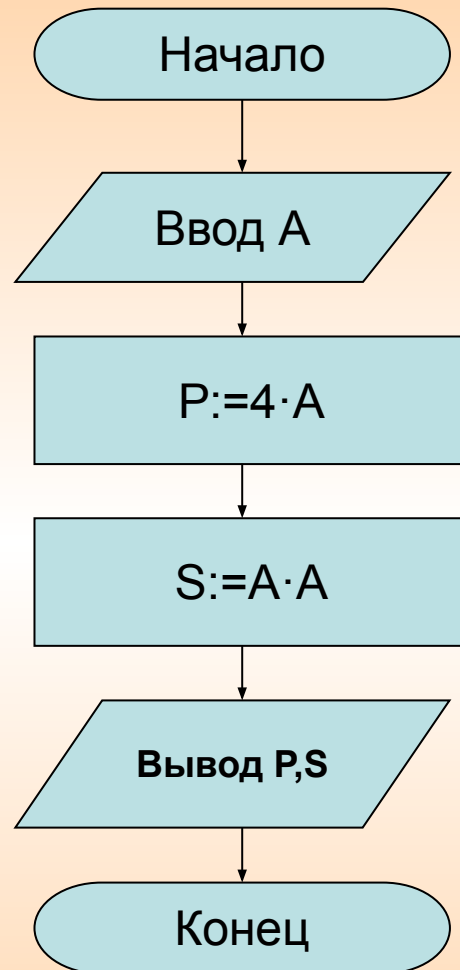
P- неизвестная переменная (периметр)

S-неизвестная переменная (площадь)

$P:=4 \cdot A$

$S:=A \cdot A$

## 2. Блок - схема



## 3. Исполнение алгоритма

A

P

S

На практике алгоритмы линейной структуры встречается крайне редко.

Чаще необходимо организовать процесс, который в зависимости от каких-либо условий проходит по той либо иной ветви алгоритма. Такой алгоритм называется **разветвляющимся.**

## 2. Разветвляющиеся алгоритмы

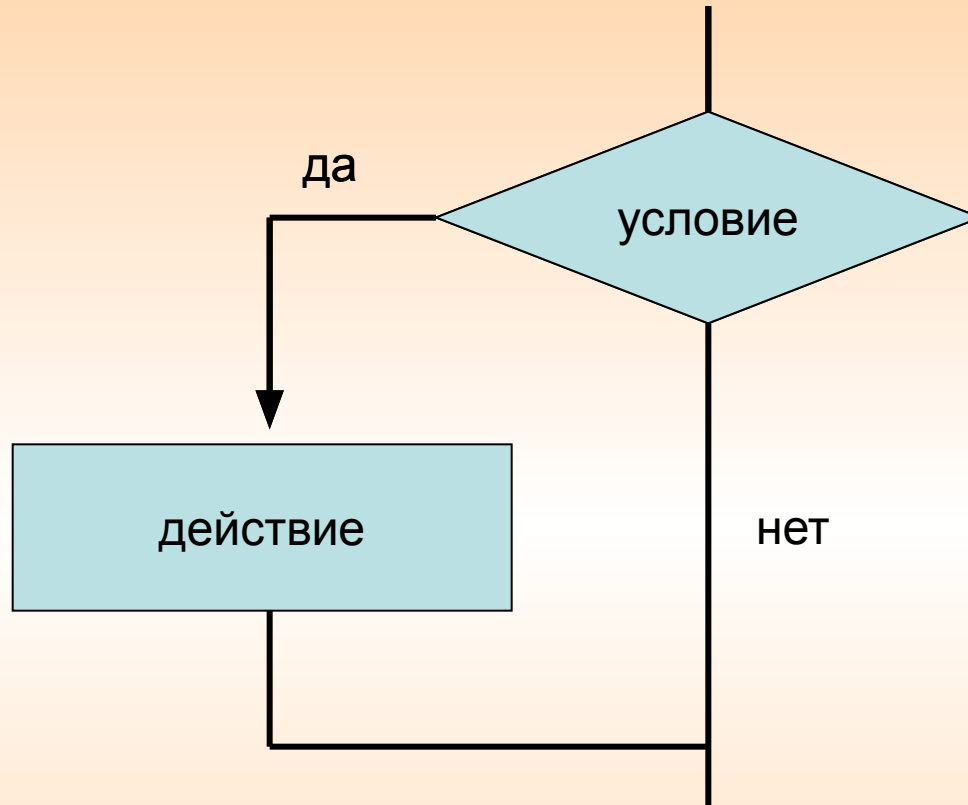
**Разветвляющийся алгоритм** - алгоритм, содержащий хотя бы одно условие, в результате которого обеспечивается переход на один из двух возможных шагов, называется разветвляющимся алгоритмом.

В блок-схемах ветвление начинается на выходах элемента "Решение", с помощью которого в алгоритме выполняется проверка какого-либо условия. Количество ветвей тем больше, чем больше проверяемых условий.

**Базовая структура ветвления**  
существует в четырех основных  
вариантах:

- **если—то** (не полное ветвление);
- **если—то—иначе** (полное ветвление);
- **выбор**;
- **выбор—иначе.**

# 1. если—то (не полное ветвление)



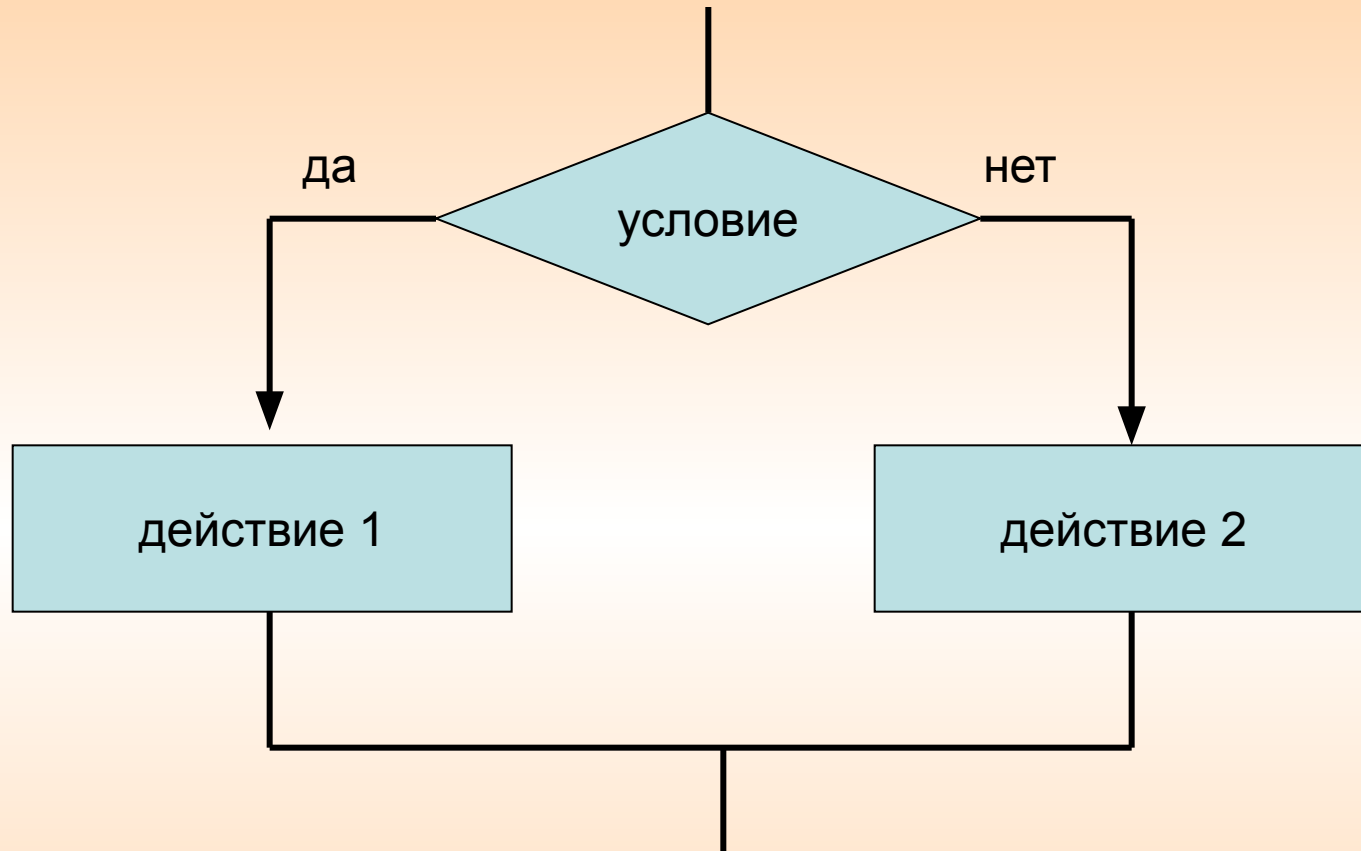
**если** условие

**то** действия

**все**

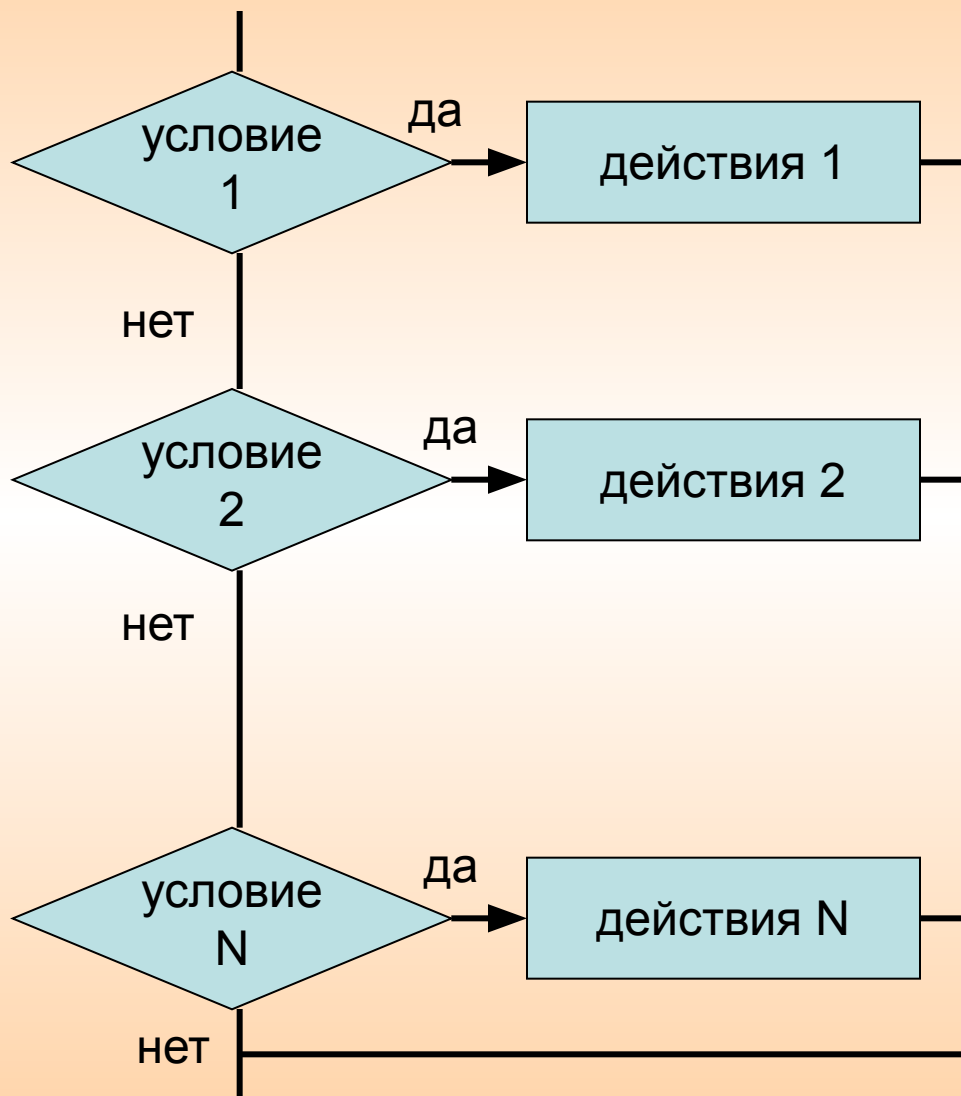


## 2. если—то—иначе (полное ветвление)



**если** условие  
**то** действия 1  
**иначе** действия 2  
**все**

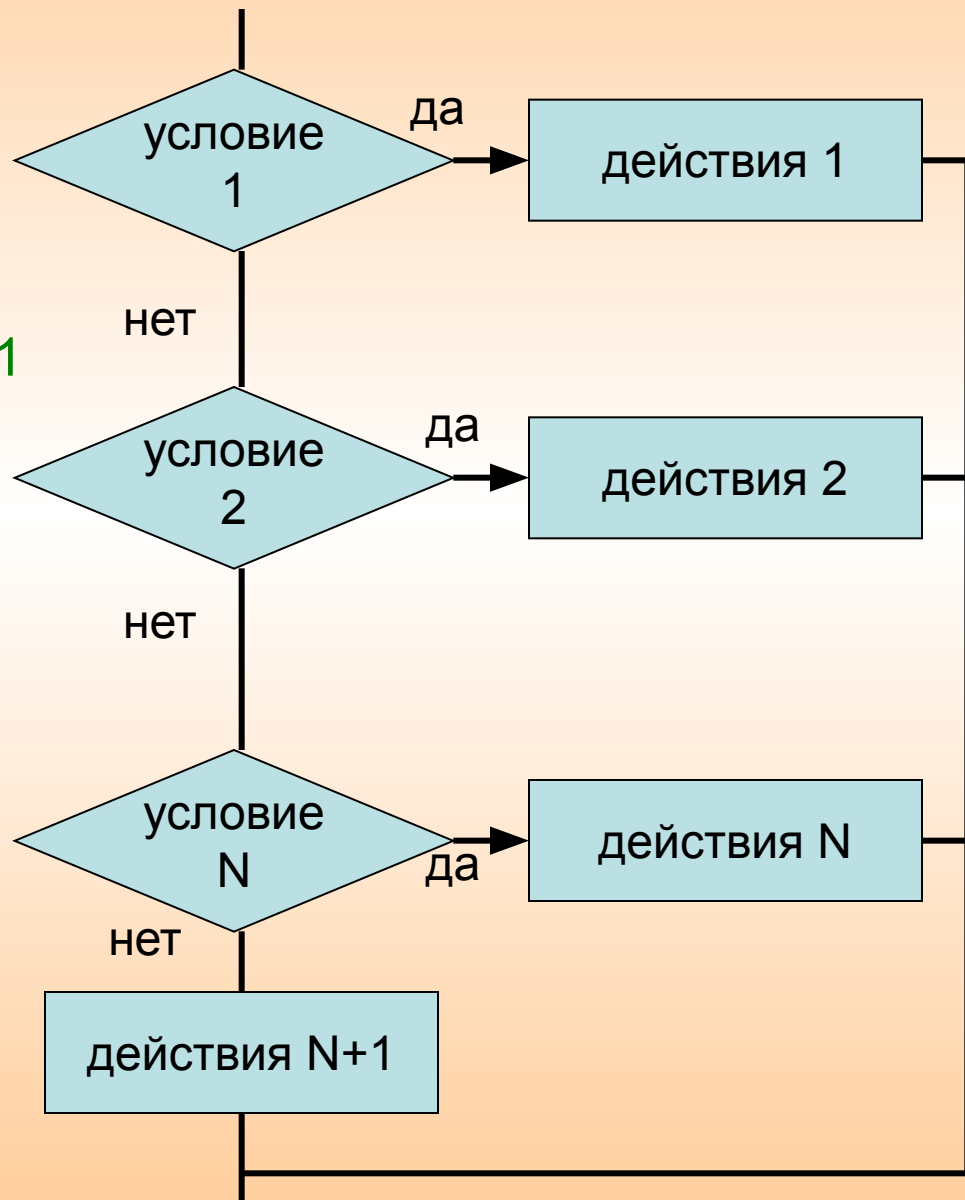
# 3. выбор



выбор при условии 1: действия 1  
при условии 2: действия 2  
.....  
при условии N: действия N  
все

# 4. выбор—иначе

**выбор** при условии 1: действия 1  
при условии 2: действия 2  
.....  
при условии N: действия N  
**иначе** действия N+1  
**все**



# Задача

Составить блок – схему алгоритма вычисления значения  $Y$  по одной из формул:

$$Y = \begin{cases} x + 2, & \text{если } x < 10 \\ x - 2, & \text{если } 10 \leq x \end{cases}$$

1. Математическое обоснование

$X$ -известная переменная

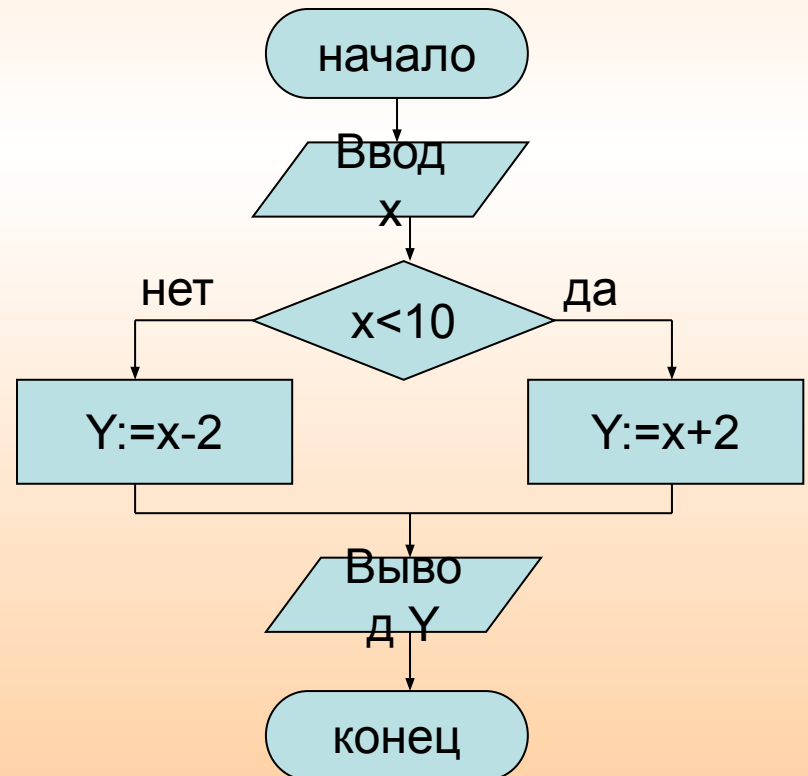
$Y$ - не известная переменная

Если  $x < 10$  тогда  $y = x + 2$  иначе  $y = x - 2$

3. Исполнение алгоритма

$x$	$y$
12	10

2 Блок-схема



# 3. Циклические алгоритмы

Часто при решении задач приходится повторять выполнение операций по одним и тем же зависимостям при различных значениях входящих в них переменных и производить многократный проход по одним и тем же участкам алгоритма.

Такие участки называются **циклами**.

Алгоритмы, содержащие циклы, называются **циклическими**.

**Использование циклов существенно сокращает объем алгоритма.**

Таким образом,

**Циклическим алгоритмом** называется алгоритм, предусматривающий многократное повторение одного и того же действия над новыми данными.

Цикл, для которого нельзя указать числа повторений, и проверка окончания цикла происходит по достижению нужного условия, называется **итерационным**.

Цикл называется **арифметическим**, если число повторений цикла известно заранее или может быть вычислено.

**Существуют три вида циклов.**

**Это: цикл “До”, цикл “Пока”, цикл “Для...”.**

Они все состоят из нескольких этапов:

- Подготовка цикла, в которую входят начальные присвоения (ПЦ);
- Тело цикла - команды повторения цикла (ТЦ);
- Подготовка данных (ПД);
- Проверка условий - обязательная часть циклов “До” и “Пока” (ПУ).

*Цикл “До”(постусловие)- это такой цикл, где тело цикла выполняется перед условием. Его лучше использовать в той циклической структуре, где заранее известно число повторений блока условия.*

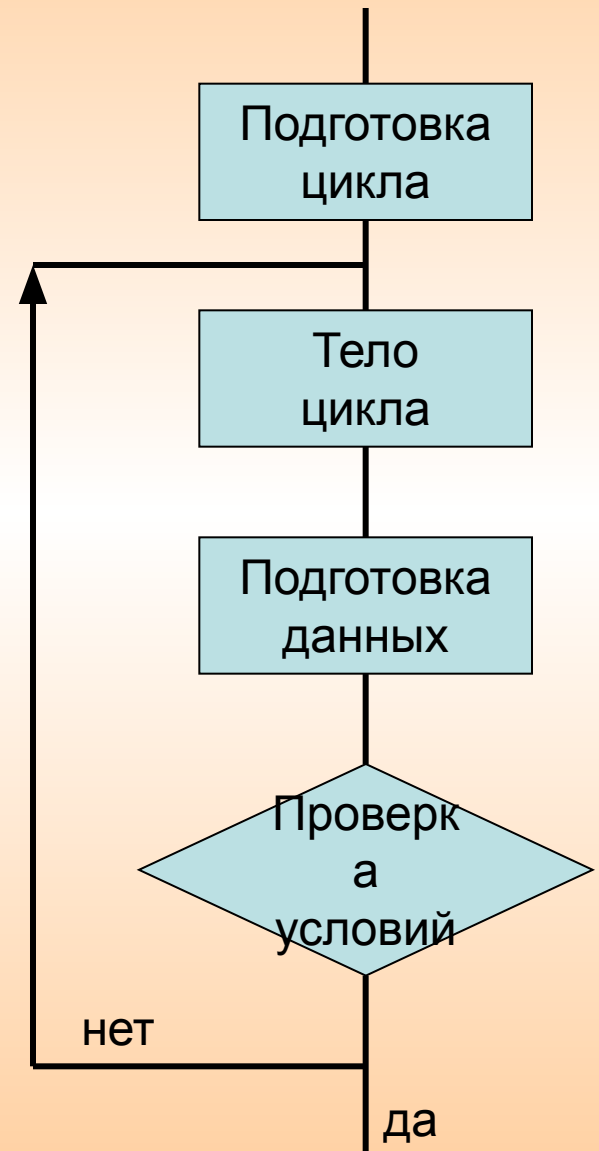
В блоке "Проверка условия" осуществляется проверка условия на прекращение цикла. Если условие справедливо (ветвь «Да»), то происходит выход из цикла, в противном случае цикл повторяется при новых значениях исходных данных.



# Цикл «ДО»

## Особенности данной структуры цикла:

- а) число повторений цикла заранее неизвестно;
- б) так как условие проверяется в конце цикла, то тело цикла выполняется как минимум один раз;
- в) возможен «бесконечный цикл», когда проверка условия не дает выхода на ветвь «Да».



*Цикл “Пока” (предусловие)- это такой цикл, где тело цикла выполняется, пока выполняются некоторые условия . Его лучше использовать там, где сразу неизвестны начальные значения цикла*

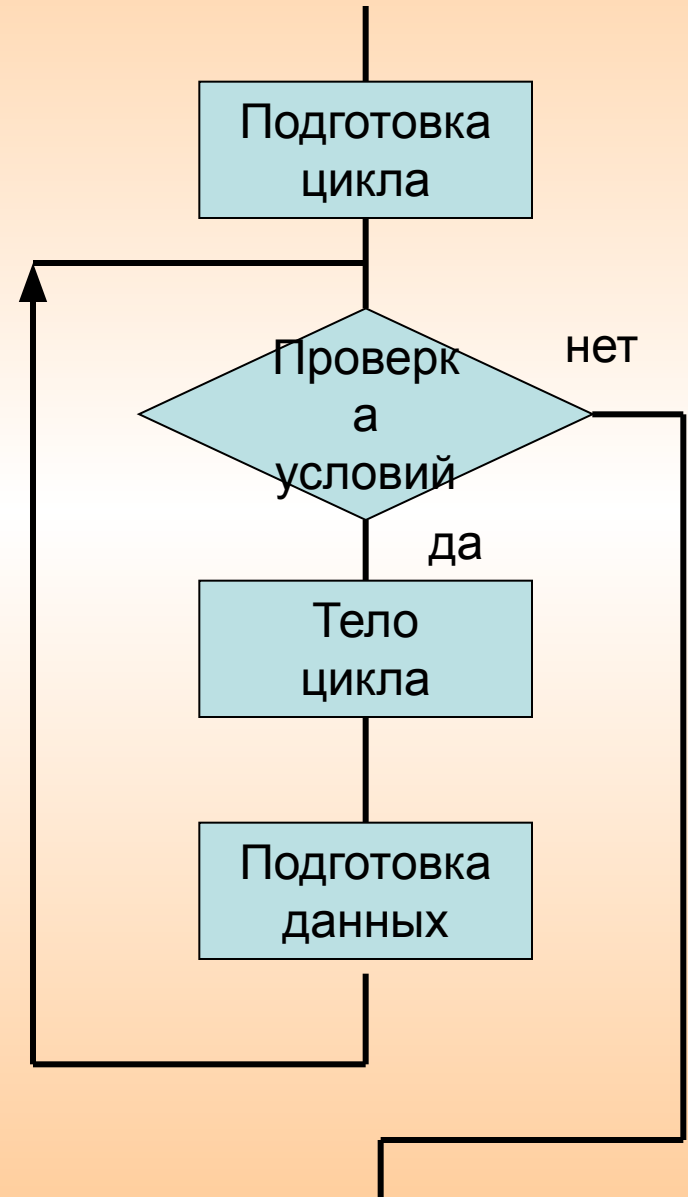
Перед выполнением операторов тела цикла осуществляется проверка условия на продолжение цикла.

Если условие справедливо (ветвь «Да»), то цикл повторяется, иначе происходит выход из цикла.

# Цикл «ПОКА»

## Особенности данной структуры цикла:

- а) число повторений цикла заранее неизвестно;
- б) если при первой же проверке условия получается "Нет", то цикл не выполняется ни разу;
- в) возможен «бесконечный цикл», когда проверка условия не дает выхода на ветвь «Нет».



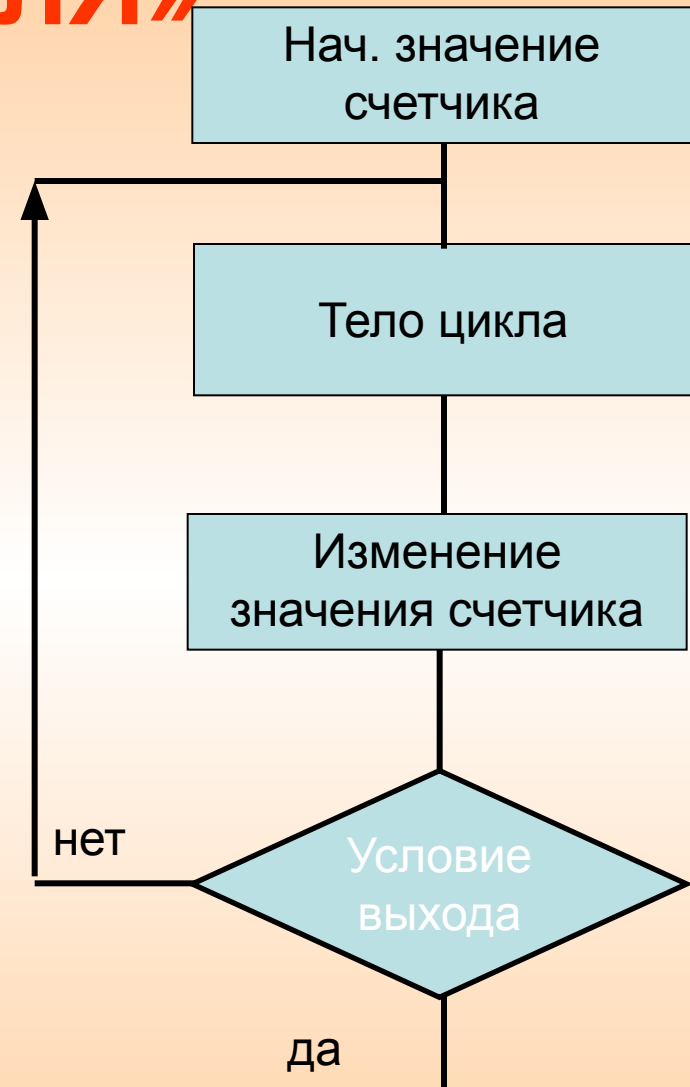
*Цикл “Для...” - это цикл с параметром, что приводит к тому, что условие не нужно. В этом случае обязательны два параметра. Это - начальное и конечное значение цикла, а так же шаг изменения.*

Тело цикла выполняется при каждом значении параметра цикла.

# Цикл «ДЛЯ»

## Особенность данной структуры цикла:

- Перед началом выполнения цикла известно количество его повторений.
- Параметр цикла, его начальное и конечное значения и шаг должны быть одного типа
- Запрещено изменять в теле цикла значения начальное, текущее и конечное для параметра
- Запрещено входить в цикл минуя блок модификации
- Если начальное значение больше конечного, то шаг - число отрицательное
- После выхода из цикла значение переменной параметра неопределенно и не может использоваться в дальнейших вычислениях
- Из цикла можно выйти не закончив его, тогда переменная параметр сохраняет свое последнее значение



# Задача

Составить блок – схему вычисления  $n!$

1. Ввод  $n$
2.  $f:=1$
3. Цикл  $i$  (начальное значение -1, конечное -  $n$ , шаг - 1)
4.  $f:=f \cdot i$
5. конец

