

# **ОСНОВЫ АЛГОРИТМИЗАЦИИ**

# **ВОПРОСЫ**

- 1. ПОНЯТИЕ АЛГОРИТМА И ЕГО СВОЙСТВА.**
- 2. СПОСОБЫ ОПИСАНИЯ АЛГОРИТМА.**
- 3. АЛГОРИТМЫ ТИПОВЫХ  
ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ.**

## Вопрос 1. ПОНЯТИЕ АЛГОРИТМА И ЕГО СВОЙСТВА

Слово «*алгоритм*» произошло от латинской формы имени среднеазиатского математика *Мухаммеда ибн Муса ал-Хорезми* (*Algoritmi*), жившего в IX в.

**АЛГОРИТМ** – точное предписание, которое определяет процесс, ведущий от исходных данных к требуемому конечному результату.

Алгоритм решения задачи разрабатывается путем разбиения ее на последовательно выполняемые шаги. При этом указываются содержание всех шагов и порядок их выполнения.

Каждый алгоритм имеет *входные и выходные данные*, которые могут отличаться структурой.

Например, алгоритм нахождения определителя квадратной матрицы:

Входные данные –  
элементы квадратной  
матрицы

3	6	5
4	5	3
1	8	2

Выходные данные –  
число



**ИСПОЛНИТЕЛЬ АЛГОРИТМА** – субъект способный выполнить действия, предписываемые алгоритмом.

Например, человек или автоматическое устройство – робот, ЭВМ, станок с ЧПУ и т.п.

**СВОЙСТВА АЛГОРИТМА** – набор  
характеристик и атрибутов, отличающих  
алгоритм от любых других предписаний.

## ОСНОВНЫЕ СВОЙСТВА АЛГОРИТМА:

**Результативность:** возможность получения результата после выполнения конечного количества операций.

**Определенность:** каждое правило алгоритма должно быть четким, однозначным и не допускать произвольной трактовки.

**Массовость:** возможность применения алгоритма для некоторого класса однотипных задач, различающихся лишь исходными данными.

**Дискретность:** алгоритм должен представлять процесс решения задачи как последовательное выполнение простых шагов.

**Понятность:** исполнитель алгоритма должен понимать, как его выполнить.

## Вопрос 2. СПОСОБЫ ОПИСАНИЯ АЛГОРИТМА

Для записи алгоритма существуют различные способы, которые отличаются по таким признакам как:

- простота,
- наглядность,
- компактность,
- степень формализации.

## **ОСНОВНЫЕ СПОСОБЫ ОПИСАНИЯ АЛГОРИТМА:**

- **словесное (вербальное) описание;**
- **формальное описание;**
- **графическое описание.**

1. При **словесном описании** алгоритм задается в произвольном изложении на естественном языке.

Алгоритм записывается в виде текста по пунктам, определяющим последовательность действий.

**Пример 1.** Записать алгоритм нахождения значения функции  $y$  для заданного значения аргумента  $x$ , где функция задана следующим выражением

$$y = \begin{cases} 6, & \text{если } x < 0 \\ 2x + 3, & \text{если } x \geq 0 \end{cases}$$

**Решение.**

1. Ввести значение аргумента  $x$ .
2. Проверить выполнение условия  $x < 0$ .
3. Если условие  $x < 0$  верно, то задать значение  $y = 6$  и перейти к шагу 5.
4. Иначе ( $x \geq 0$ ) найти значение  $y = 2x + 3$  и перейти к шагу 5.
5. Вывести результат  $y$ .
6. Закончить вычисления.

Словесный способ не имеет широкого распространения, так как описания:

- строго не формализуемы;
- страдают многословностью записей;
- могут допускать неоднозначность толкования отдельных предписаний.

2. При ***формальном описании*** алгоритм записывается на формализованном (формальном) языке, который характеризуется точными правилами построения и понимания выражений.

В ходе построения формальных языков были созданы алгоритмические языки.

**Алгоритмический язык** — это система наименований для формальной записи алгоритмов, которые предназначены для некоторого исполнителя. В состав алгоритмического языка входят команды, операторы, служебные слова и символы.

(Он обладает своим синтаксисом и семантикой).

## **Пример 2.** Алгоритм для примера 1 на одном из школьных алгоритмических языков:

```
алг Функция (вещ X, Y)
    арг X
    рез Y
нач
    ввод X
    если X < 0
        то Y = 6
    иначе Y = 2X + 3
    все
    вывод Y
кон
```

Недостатки алгоритмических языков: алгоритмы не всегда наглядны, часто объемны и громоздки.

При реализации алгоритма на ПК его нужно записать на одном из языков программирования в виде исходной программы.

(Языки программирования обладают высокой степенью формализации по сравнению с алгоритмическими языками).

**Пример 3.** Описание алгоритма из примера 1 средствами языка QBasic:

**REM Вычисление функции**

**INPUT x**

**IF x>0 THEN y=0 ELSE y=2\*x+3**

**PRINT y**

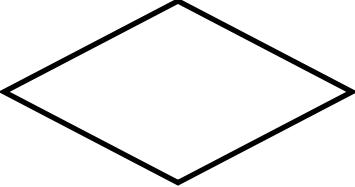
**END**

3. При *графическом описании* алгоритм изображается в виде *блок – схемы*, состоящей из взаимосвязанной последовательности геометрических фигур (блоков), каждая из которых подразумевает выполнение определенного действия (ввод исходных данных, проверка условия и т.д.).

Условные обозначения блоков выполняются  
в соответствии со стандартом –  
**ГОСТ 19.701-90 ЕСПД**  
«Схемы алгоритмов, программ, данных  
и систем».

# Основные блоки, используемые в блок-схемах

Наименование	Изображение операции	Функция
ПРОЦЕСС		Выполнение операции, в результате которой изменяются значение, форма представления или расположения данных <i>(Операция присваивания)</i>

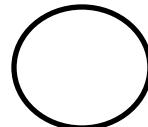
<b>РЕШЕНИЕ</b>		Выбор направления алгоритма в зависимости от заданных условий <i>(Условный переход)</i>
<b>ВВОД-ВЫВОД</b>		Преобразование данных в форму, пригодную для обработки ( <i>Ввод</i> ) или отображения результатов обработки ( <i>Выход</i> )
<b>ТЕРМИНАТОР</b>		Начало, конец, прерывание процесса обработки данных <i>(Начало, Останов.)</i>

**ДОКУМЕНТ**



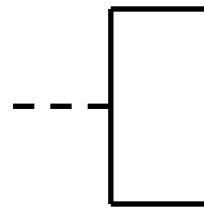
Вывод данных на  
бумажный носитель  
(*Печать*)

**СОЕДИНИТЕЛЬ**



Указание связи  
между прерванными  
линиями,  
соединяющими  
блоки (элементы  
схемы)

**КОММЕНТАРИЙ**

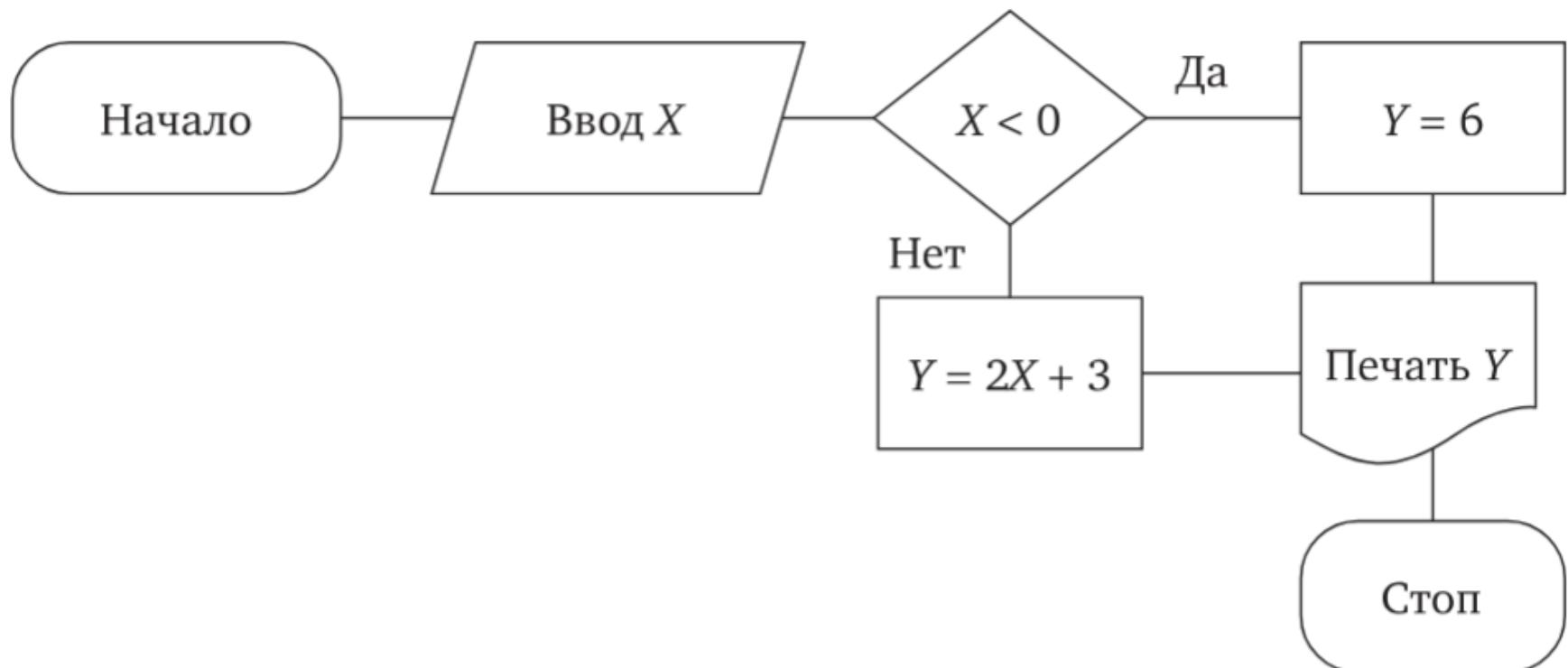


Связь между  
элементом схемы и  
пояснением

Блоки соединяются между собой линиями переходов. Порядок выполнения действий указывается стрелками.

Блоки нумеруются сверху–вниз и слева–направо. Такая последовательность исполнения блоков называется *естественной*.

## Пример 4. Блок-схема для алгоритма примера 1.



### **3. АЛГОРИТМЫ ТИПОВЫХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ**

Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур:

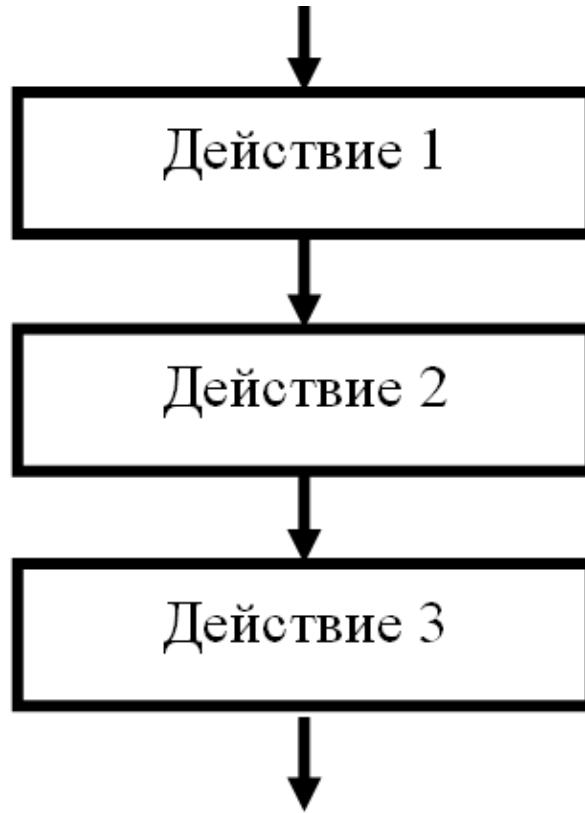
- ✓ **линейная,**
- ✓ **разветвляющаяся,**
- ✓ **циклическая.**

**Линейным** называется алгоритм, в котором операции выполняются последовательно, в порядке их записи.

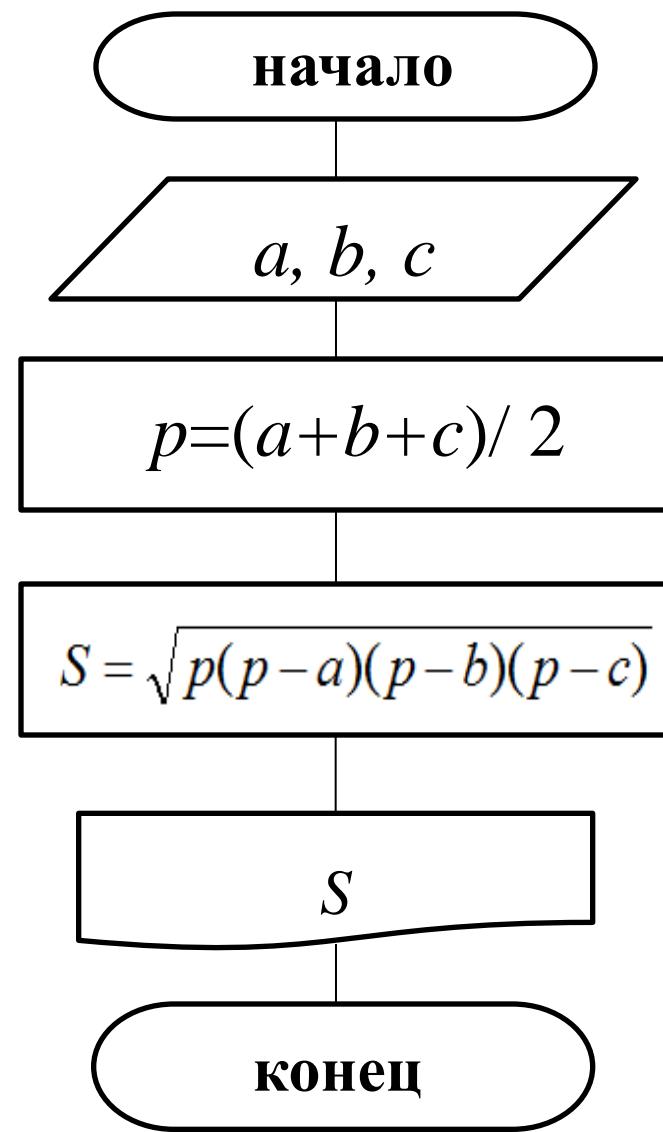
Каждая операция является самостоятельной, независимой от каких-либо условий.

На схеме блоки, отображающие эти операции, располагаются в линейной последовательности.

# Линейная структура алгоритма



**Пример 3.** Блок-схема алгоритма для вычисления площади треугольника по формуле Герона.



**Разветвляющимся** называется алгоритм, в котором действия (операции) выполняются в той или иной последовательности в зависимости от поставленного условия.

# Алгоритмическая структура «ветвление»

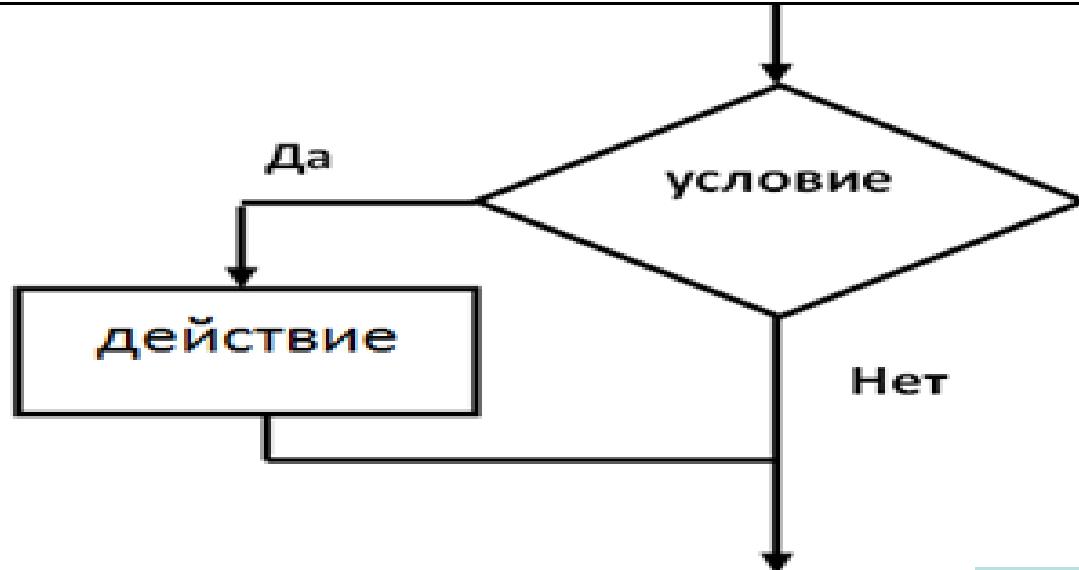


Рис. 1 – Неполное ветвление

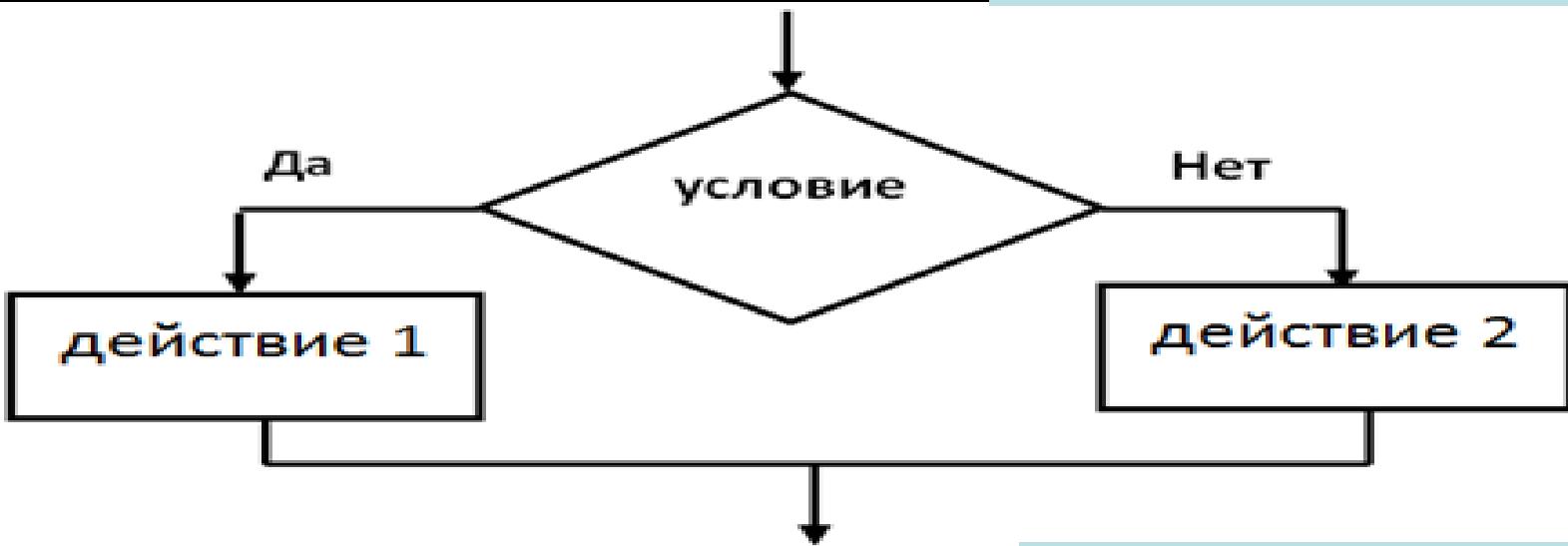
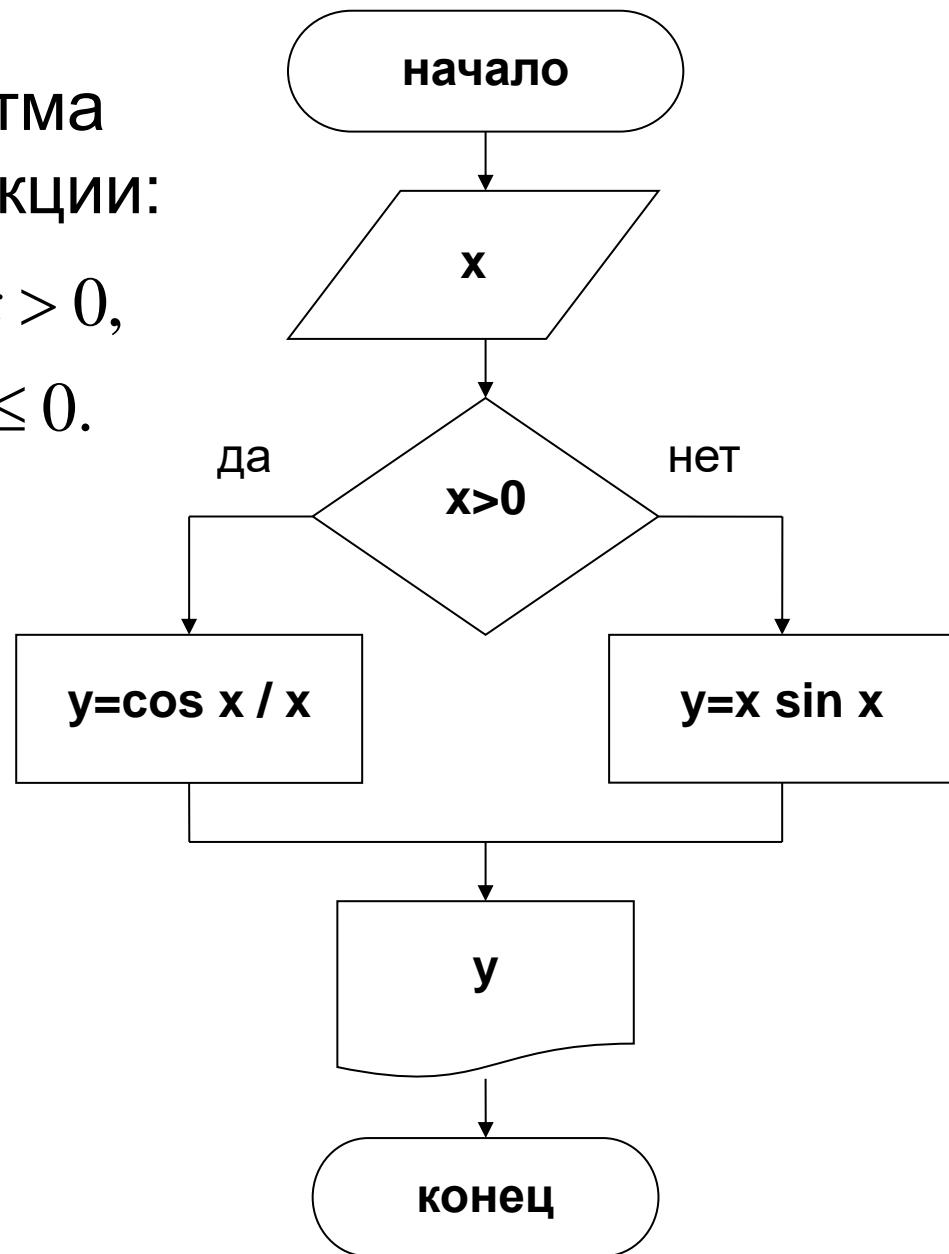


Рис. 2 – Полное ветвление

## Пример 5.

Блок-схема алгоритма  
для вычисления функции:

$$y = \begin{cases} \cos x / x, & \text{если } x > 0, \\ x \sin x, & \text{если } x \leq 0. \end{cases}$$



**Циклическим** называется алгоритм, в котором предусмотрено неоднократное выполнение одной и той же последовательности действий при изменяющихся исходных данных.

Многократно выполняемую последовательность действий называют **телом цикла**.

Циклы могут организовываться по-разному.

## Цикл с параметром (со счётчиком)

*Параметр цикла* – это переменная, которая при каждом новом входе в цикл принимает новое значение и управляет работой цикла.

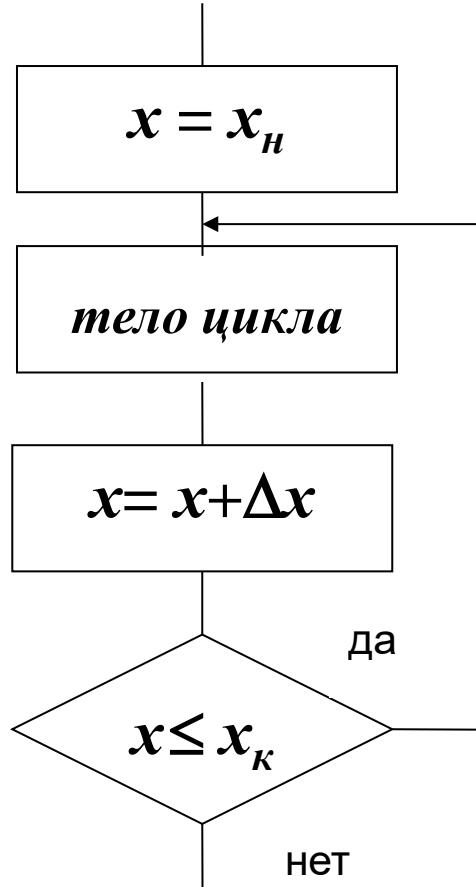
В организации цикла с параметром выделяют следующие этапы:

- Задается начальное значение *параметра цикла* ( $x_n$ );
- Выполняется *тело цикла*;
- Значение параметра изменяется на величину, называемую *шагом* ( $\Delta x$ );
- Проверяется *условие продолжения цикла* – цикл выполняется до тех пор, пока величина параметра цикла не превышает заданного конечного значения ( $x_k$ ).

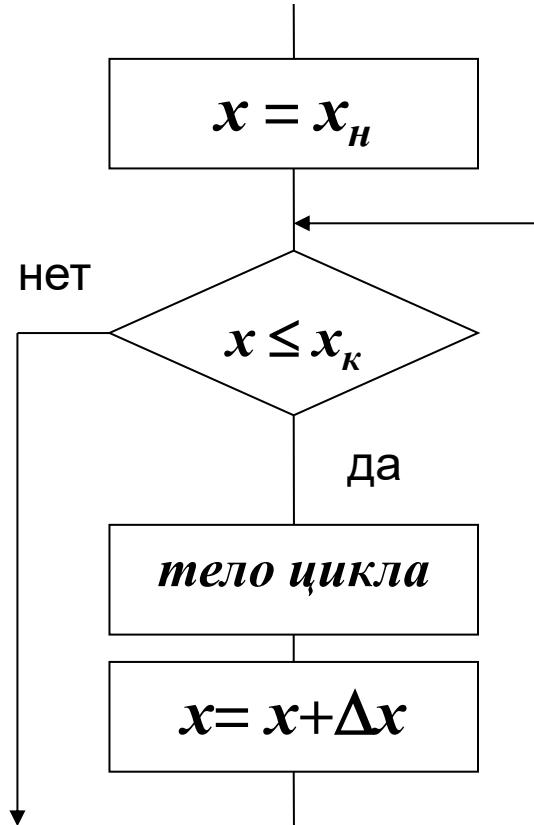
В зависимости от расположения условия продолжения цикла различают:

- цикл с постусловием (типа До);
- цикл с предусловием (типа Пока).

# Алгоритмическая структура «Цикл с постусловием»



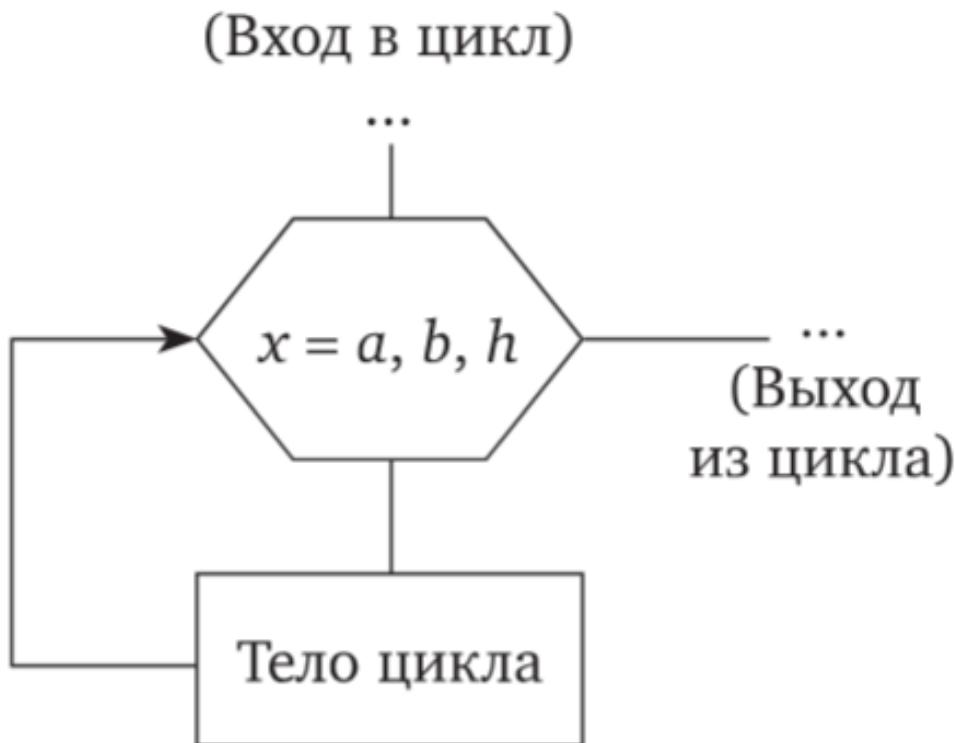
# Алгоритмическая структура «Цикл с предусловием»



Для обозначения циклического процесса можно использовать специальный блок «подготовка».

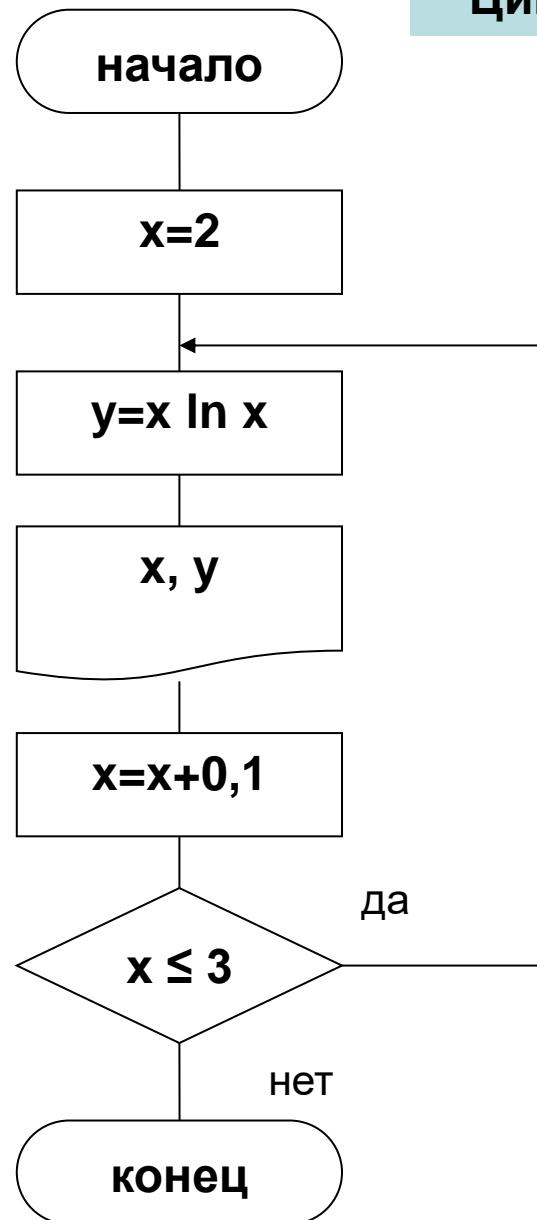
В блоке указываются характеристики параметра цикла (начальное, конечное значения и шаг изменения в формате  $x = x_{\text{Н}}, x_{\text{К}}, \Delta x$ ).

## Типовая схема цикла с блоком «подготовка»

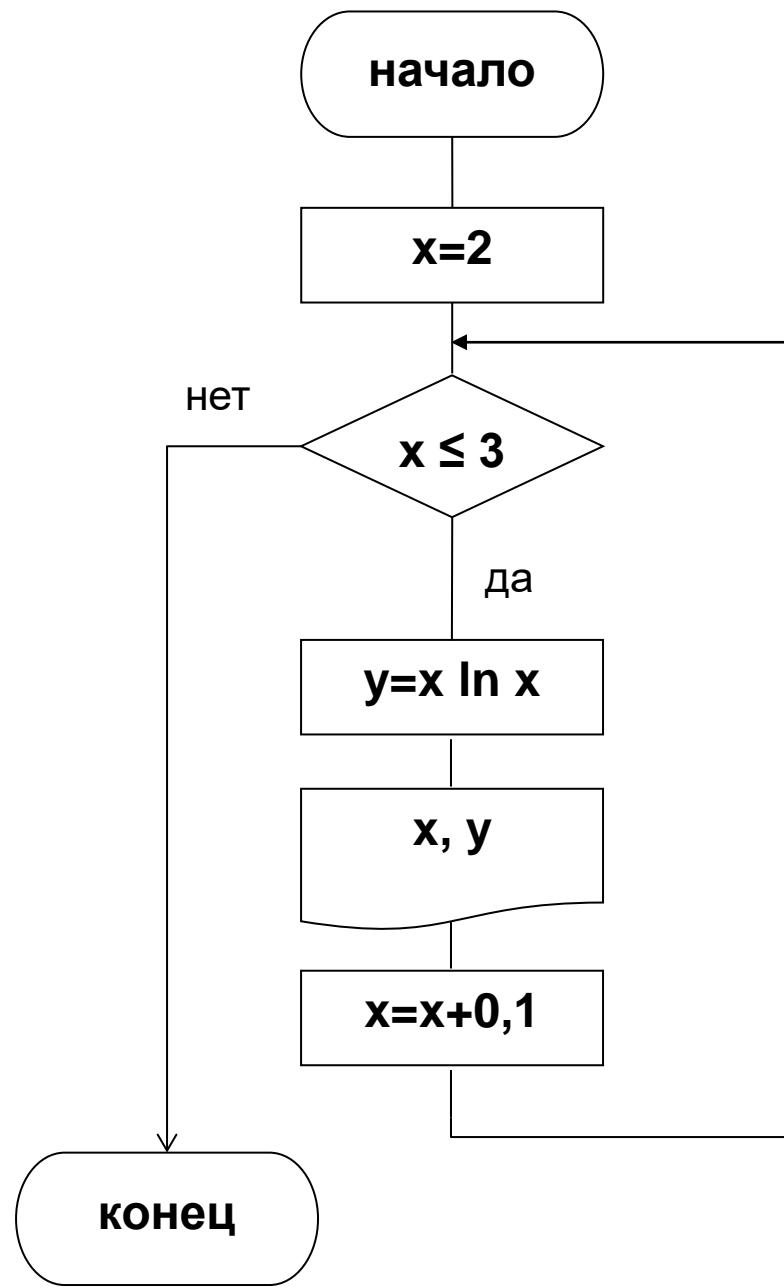


## Пример 6.

Блок-схема алгоритма для вычисления функции  $y = x \ln x$  на интервале  $[2; 3]$  с шагом 0,1.



## Цикл с предусловием



## Цикл с блоком «подготовка»

