

# Тема: Базовые алгоритмические конструкции

Раздел: Алгоритмизация и программирование



# ПЛАН:

1. Алгоритмы линейной структуры.
  2. Алгоритмы разветвляющейся структуры.
  3. Алгоритмы циклической структуры.
-

# Базовые алгоритмические конструкции

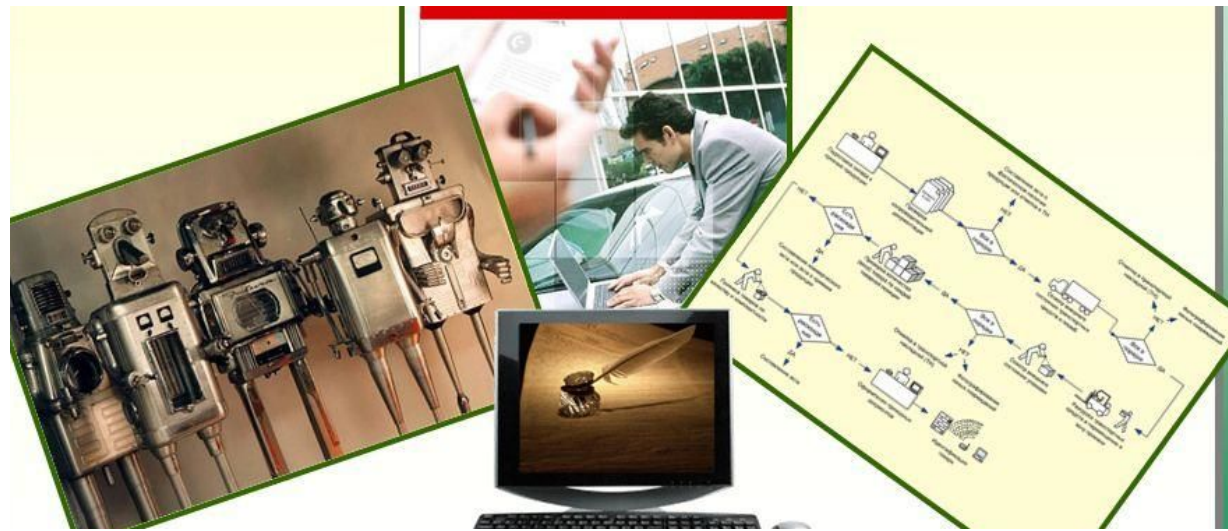
Для повышения эффективности применения компьютера как инструмента решения задач необходимо освоение основной фундаментальной концепции подхода к использованию цифровых вычислительных средств. В информатике таким фундаментом является алгоритмизация задач.



**Исполнитель алгоритма** — это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.

**Исполнителя характеризуют:**

- система команд;
- среда;
- отказы.



## Система команд исполнителя

- **совокупность команд**, которые данный исполнитель может выполнять;
- алгоритм описывается в командах исполнителя, который будет его реализовывать.

## Среда исполнителя

- **объекты**, над которыми исполнитель может совершать действия;
- исходные данные и результаты любого алгоритма принадлежат среде исполнителя, для которого предназначен алгоритм.

## Отказ

- возникает в случае, если команда вызывается при недопустимом для нее состоянии среды.
- для каждой команды должны быть заданы **условия применимости** (в каких состояниях среды может быть выполнена команда) и описаны **результаты выполнения команды**.

# Базовые алгоритмические конструкции

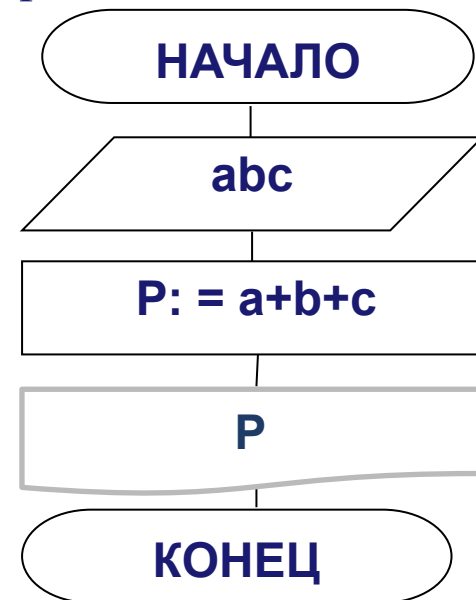


# 1. Алгоритмы линейной структуры

Алгоритм линейной структуры - алгоритм, в котором блоки выполняются последовательно друг за другом, в порядке, заданном схемой. Такой порядок выполнения называется естественным.

*Пример.*

*Вычислить периметр треугольника со сторонами  $a, b, c$*



Основу линейного алгоритма составляют три алгоритмические конструкции : *операция ввода* , *операция присваивания* , *операция вывода*.

**Операция присваивания** используется для задания значения некоторой переменной, имеет вид

**переменная:=значение**

При присваивании сначала вычисляется значение справа от знака «присвоить» **:=** , затем это значение записывается в переменную

Составить алгоритм вычисления функции вида  $y=x+3z$ , для заданных значений  $x$  и  $z$ .

НАЧАЛО

ВВОД  $x, z$

$y := x + 3 * z$

ВЫВОД  $y$

КОНЕЦ



# Алгоритмы линейной структуры

**Требования к именам (идентификаторам) переменных:** имена могут включать латинские буквы, цифры, **всегда начинается с буквы.**

Например, возможен объект с именем A1, но не 1A.

Переменные должны иметь определенный **тип данных.**

Справа от знака "присвоить" может находиться не только переменная или константа, но и **арифметическое выражение (формула).**

$S := v * t$

$A := 0$

Арифметические выражения строятся из **операндов**, которыми могут быть **константы, переменные и стандартные функции.**

---

# Алгоритмы линейной структуры

В выражение могут входить *арифметические операции* и круглые скобки. В большинстве языков определено **6 арифметических операций**, перечислим их в соответствии с *приоритетом*, операции с одинаковым приоритетом равноправны между собой и выполняются слева направо, как и в математике.

Приоритет	Обозначение операции	Описание операции
1	*	Умножение
	/	Деление
	div	Деление двух целых значений с отбрасыванием остатка
	mod	Взятие остатка от деления двух целых значений
2	+	Сложение
	-	Вычитание

---

# Алгоритмы линейной структуры

- ❖ При необходимости изменить обычное старшинство операций в записи выражения используются *дополнительные круглые скобки*.

- ❖ Запись выражения

$$y = \frac{a+b}{2}$$

Правильная  
 $y := (a+b)/2$

Запись неверна  
 $y := a+b/2$

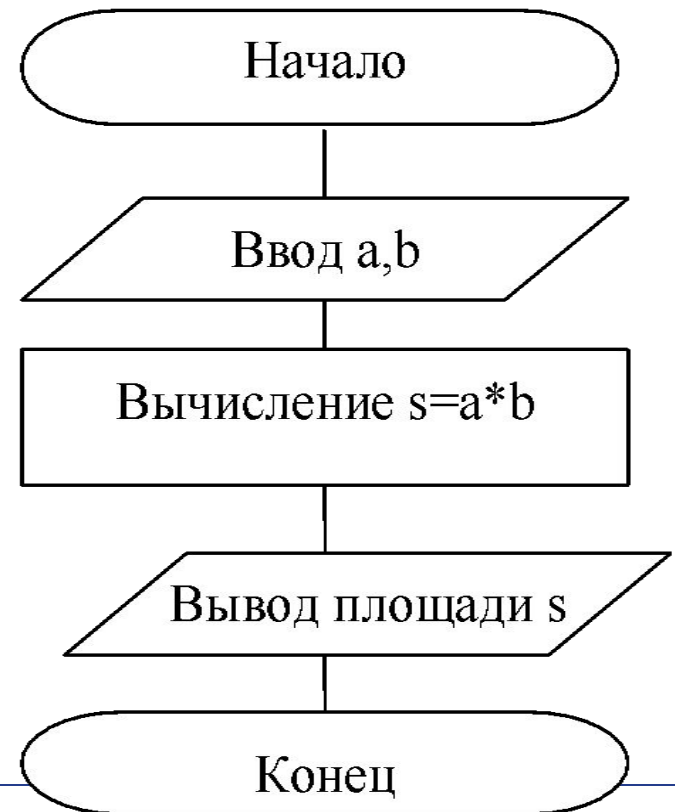
$y := 2012;$   
 $c := y \text{ div } 100;$   
 $n := y \text{ mod } 100;$

переменная  $c = 20,$   
 $n = 12$

# Пример

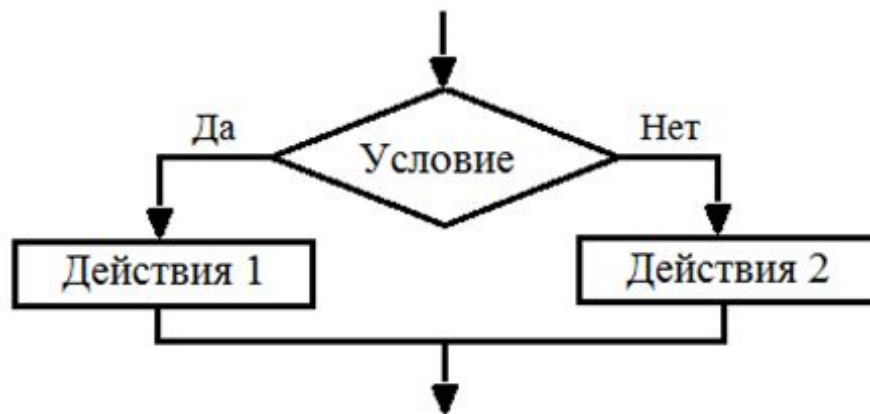
Составить алгоритм вычисления площади прямоугольника  $s$  по известным длинам сторон  $a$ ,  $b$ .

- ❖ Исходные данные:  $a$  - длина прямоугольника,  $b$  - ширина прямоугольника.
- ❖ Выходные данные:  $s$  - площадь.
- ❖  $S=a*b$  математическая модель

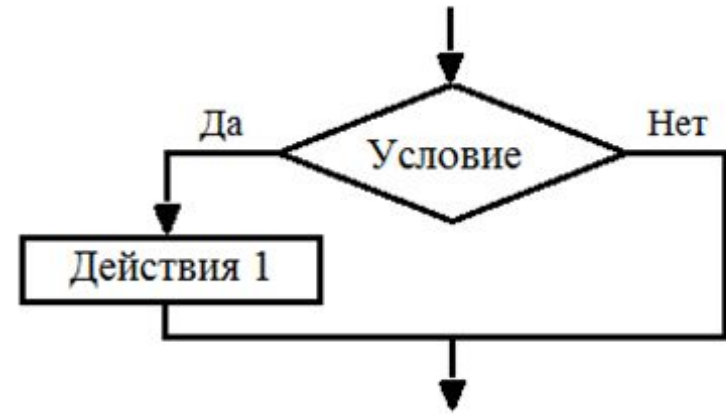


## 2. Алгоритмы разветвляющейся структуры

**Разветвляющимся** называется алгоритм, в котором действие выполняется по одной из возможных ветвей решения задачи, в зависимости от выполнения условий.



Полная форма

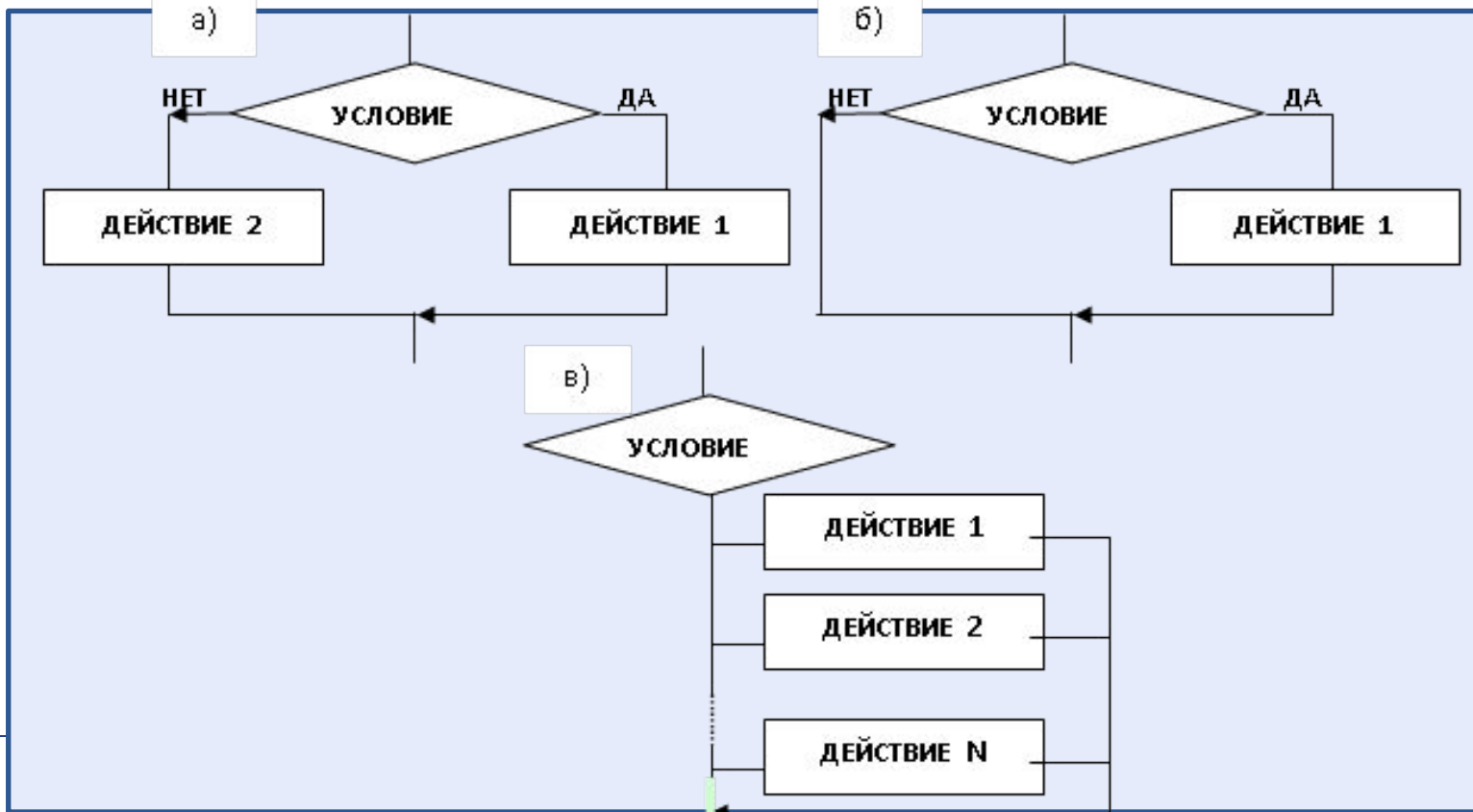


Сокращенная форма

# Алгоритмы разветвляющейся структуры

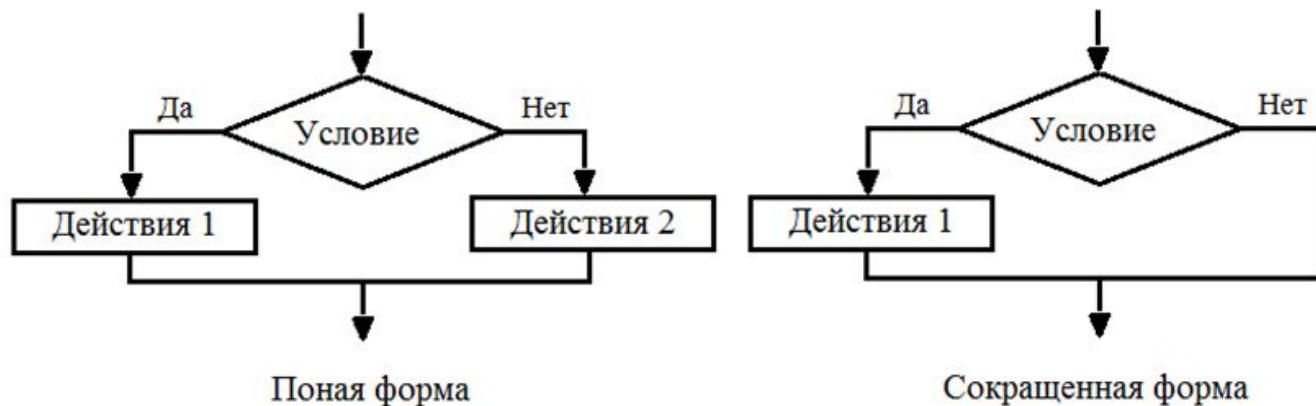
Структура «ветвление» существует в трёх основных вариантах:

**если-то-иначе** (рисунок 3.а); **если-то** (рисунок 3.б);  
**выбор-иначе** (рисунок 3.в).



# Алгоритмы ветвления

- ❖ **Условие** – логическое выражение, которое может быть истинным или ложным.
- ❖ В качестве **условия** в разветвляющемся алгоритме может быть использовано любое понятное исполнителю утверждение, которое может быть выражено как словами, так и формулой.
- ❖ Алгоритм ветвления состоит из **условия** и **последовательностей команд**.



# Виды условий

**Простое условие** – это условие, в котором используются переменные и операции сравнения

- $>$  - «больше»,
- $<$  - «меньше»,
- $=$  - «равно»,
- $<>$  - «не равно»,
- $\geq$  - «больше или равно»
- $\leq$  - «меньше или равно».

- $A \geq 0$
- $A \leq 9$
- $A < B$

**Составное условие** – это несколько простых условий, соединённых логическими операциями

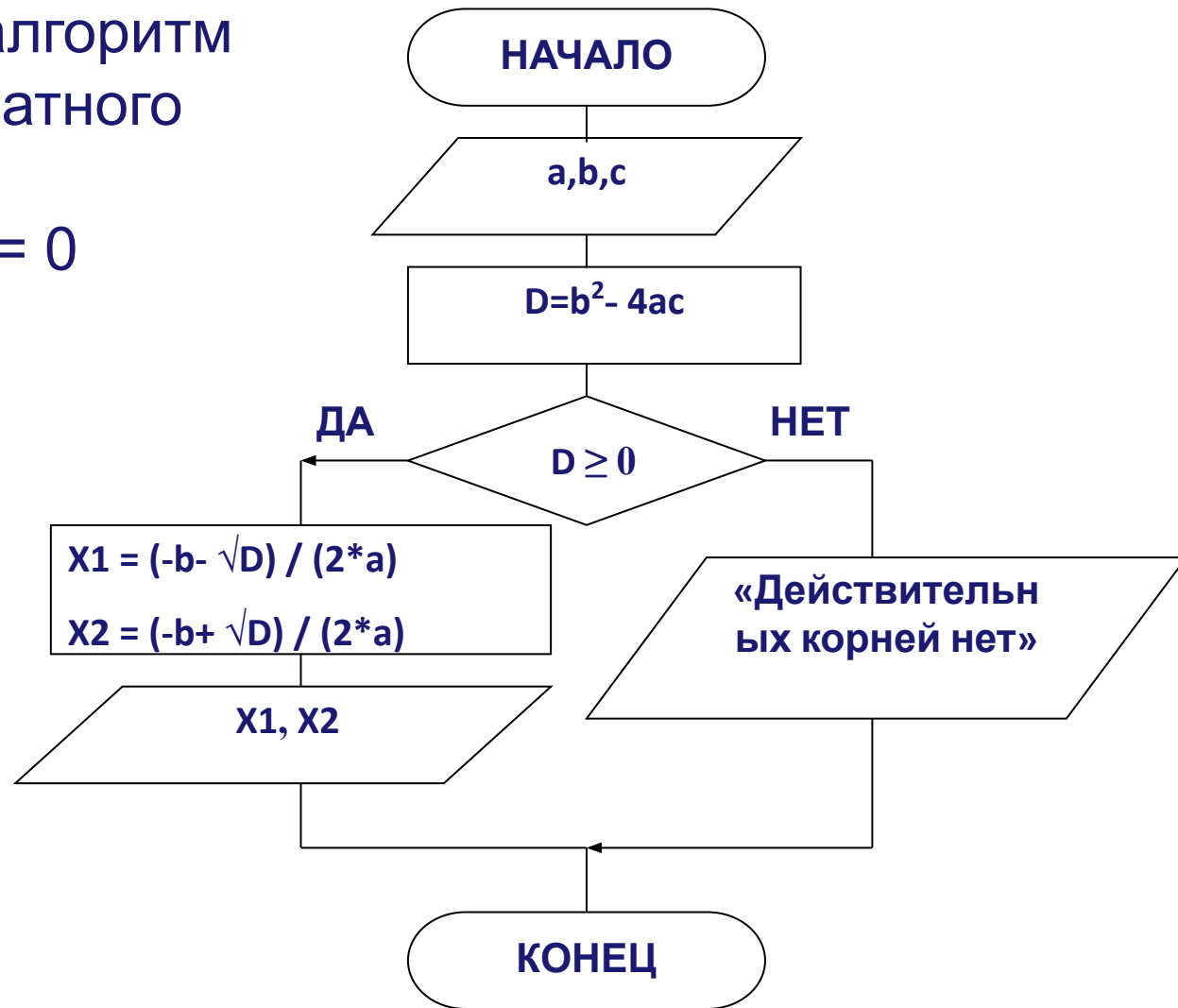
- not – «нет»,
- or – «или»,
- and – «и».
- Знаки логических операций называют логическими связками.

$(A \geq 10) \text{ и } (A \leq 99)$



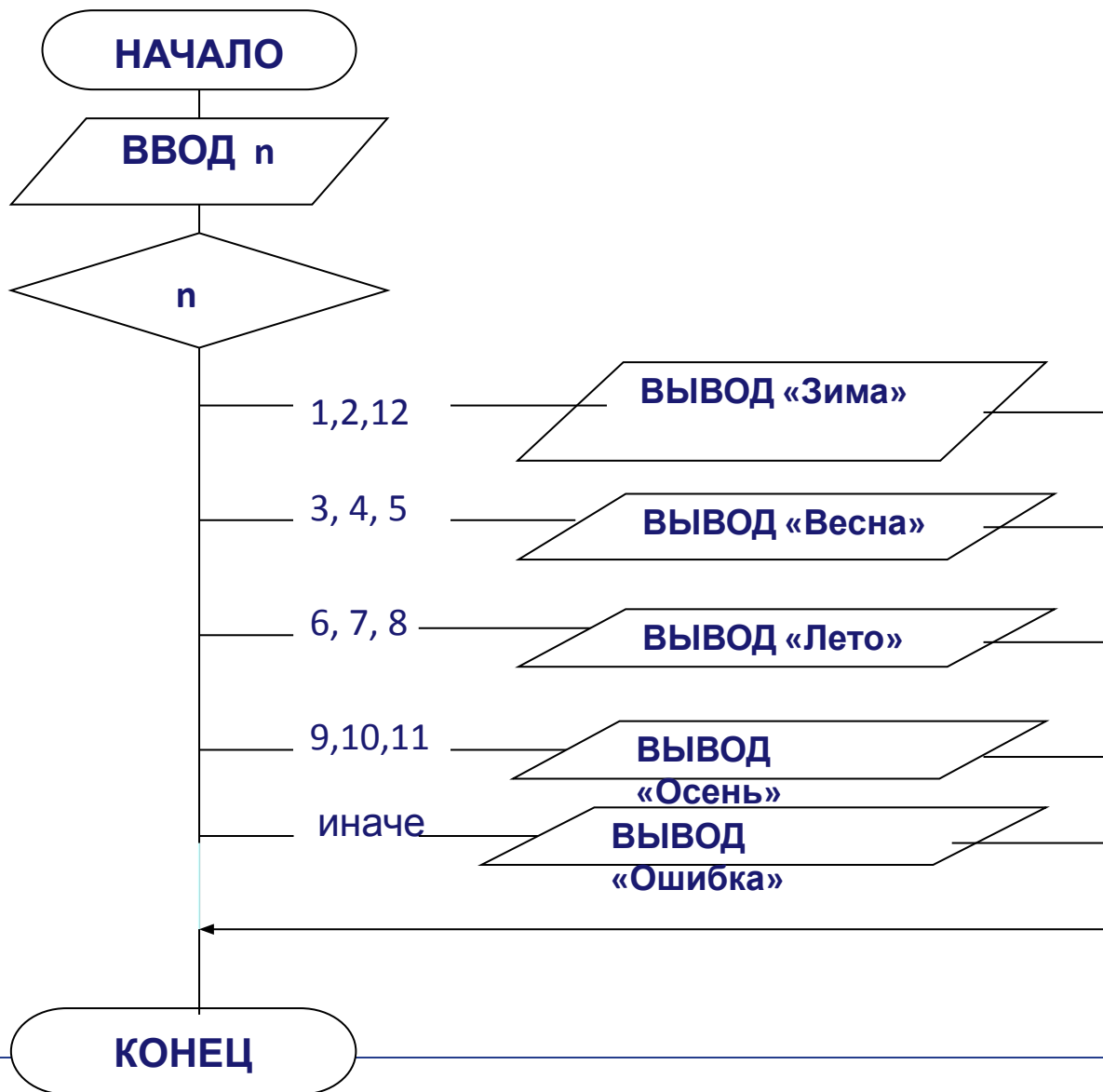
# Пример алгоритма ветвления

Составить алгоритм  
решения квадратного  
уравнения  
 $ax^2 + bx + c = 0$



# Алгоритмы ветвления

Составить алгоритм, который по номеру месяца  $n$  выводит название времени года, соответствующего данному месяцу



# 3. Алгоритмы циклической структуры

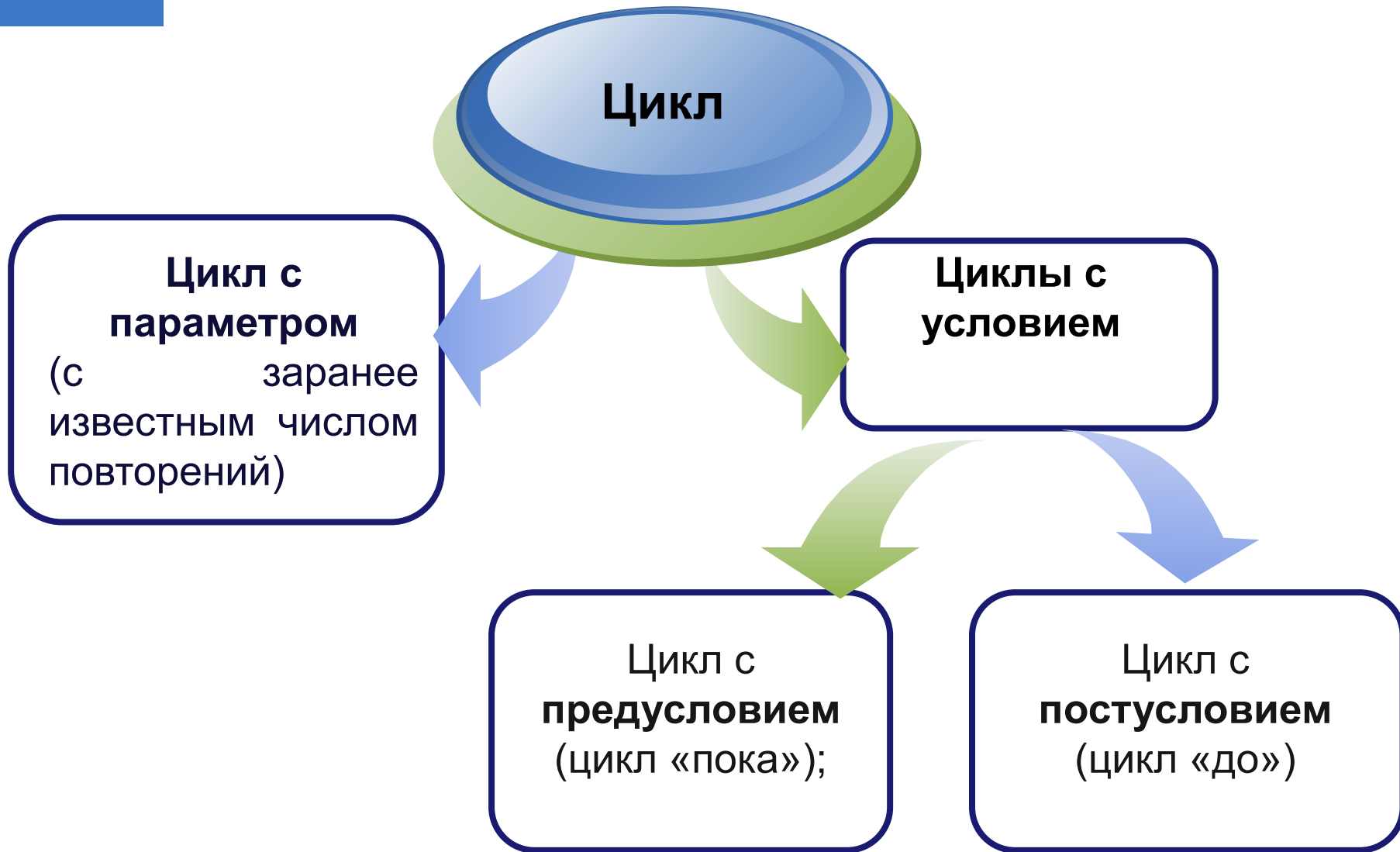
**Базовая структура «цикл»** обеспечивает многократное выполнение некоторой совокупности действий.

Повторяющаяся совокупность действий называется – **телом цикла**.

Величина, с которой связано многократное выполнение тела цикла называется – **параметром цикла**. Параметр цикла имеет **начальное** и **конечное** значения.

**Шаг цикла** – величина на которую изменяется значение параметра цикла при каждом выполнении цикла.

# Виды циклов



# Цикл с параметром

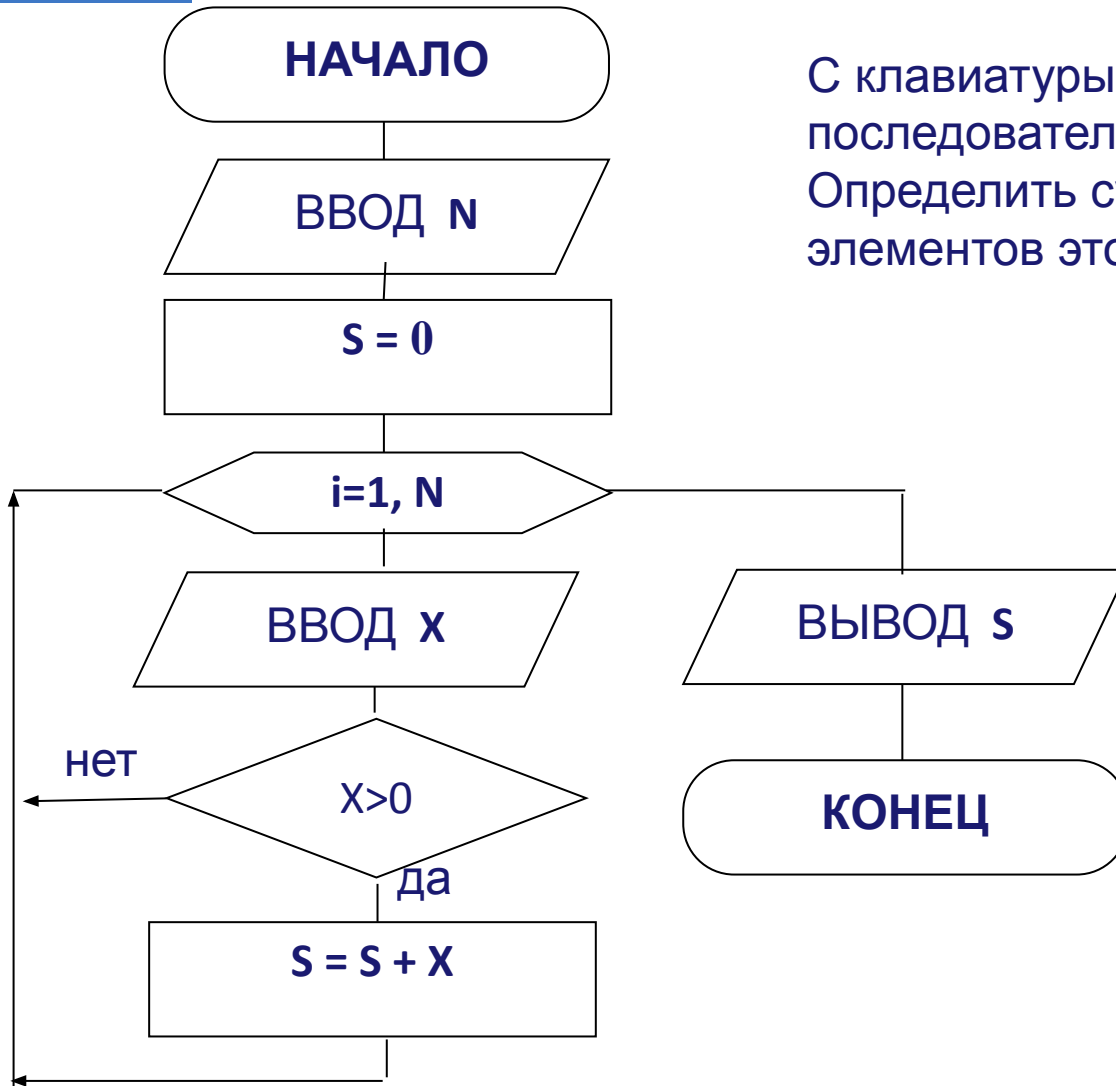


## *Работа цикла*

- Параметру цикла **P** присваивается начальное значение **N** и происходит выполнение тела цикла.
- Далее значение параметра цикла увеличивается на величину шага **H** и проверяется условие: (текущее значение параметра цикла должно быть меньше конечного **K** значения или равно ему  $P \leq K$ ).
- Цикл будет повторяться до тех пор, пока это условие истинно.
- Как только **P** станет больше **K** ( $P > K$ ) произойдет выход из цикла

# Цикл с параметром

С клавиатуры вводится последовательность из  $N$  чисел. Определить сумму положительных элементов этой последовательности



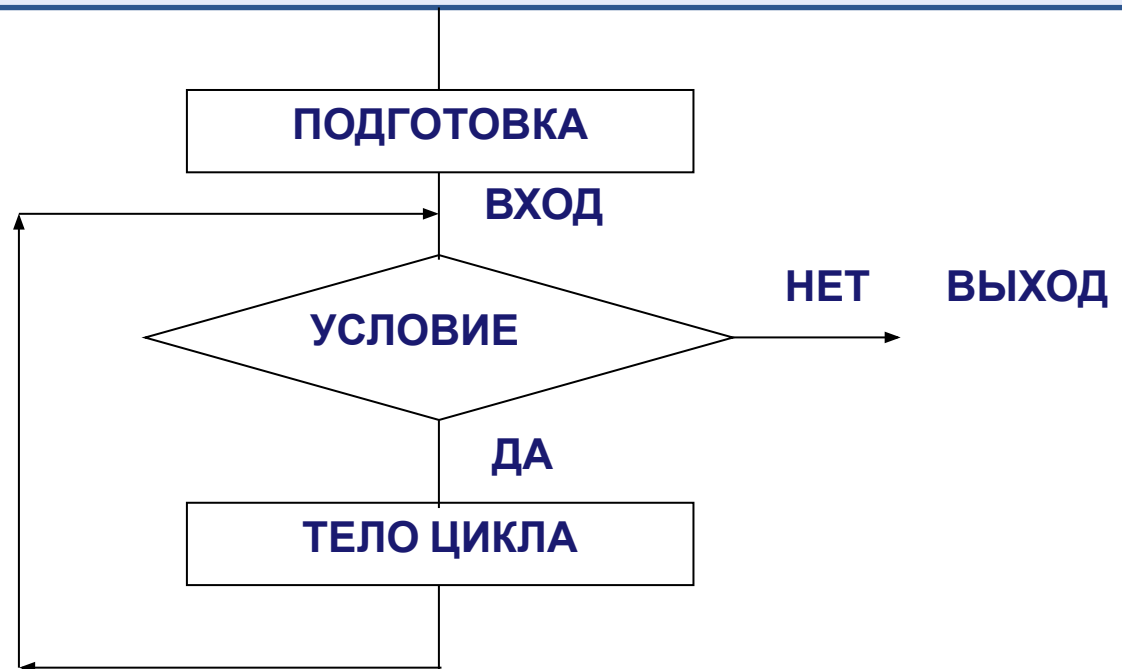
# Цикл с предусловием

Проверка условия продолжения цикла проводится до выполнения действий цикла. В циклах с условием, как правило, выполняется **подготовительный процесс**:

- задаются начальное  $n$  и конечное  $k$  значения параметра цикла  $p$
- задается величина шага  $h$

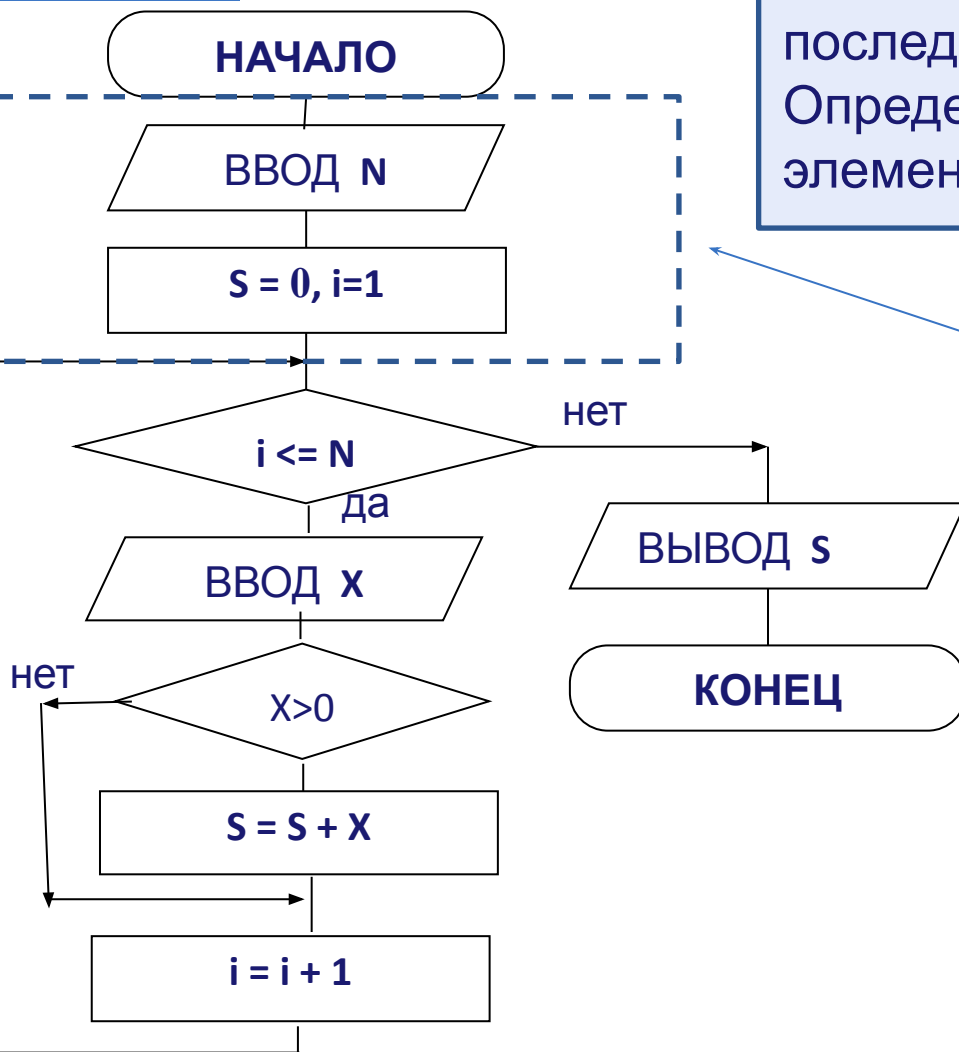
В теле цикла значение параметра цикла увеличивается на величину шага  $h$

Цикл с предусловием может не выполняться ни разу, если условие сразу же окажется ложным.



# Цикл с предусловием

С клавиатуры вводится последовательность из  $N$  чисел. Определить сумму положительных элементов этой последовательности

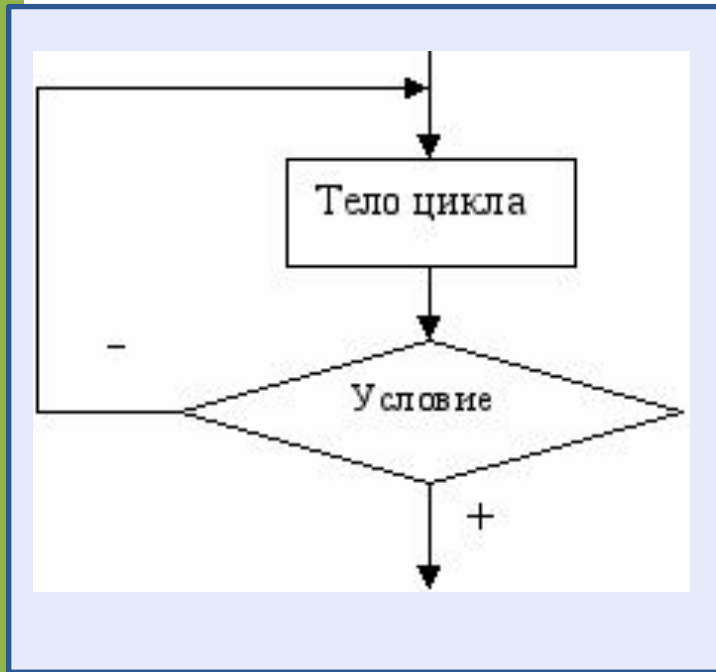


**Этап подготовки** в данной схеме включает в себя: ввод конечного значения параметра цикла  $N$ , задание начального значения  $i$ , обнуление суммы  $S$ .

**Цикл** начинается с проверки условия выполнения цикла. В данном случае цикл должен выполняться пока значение параметра  $i \leq N$ . В теле цикла вычисляется значение суммы, а далее производится изменение параметра цикла на величину шага равную 1. Как только условие станет ложным, производится выход из цикла и вывод результата



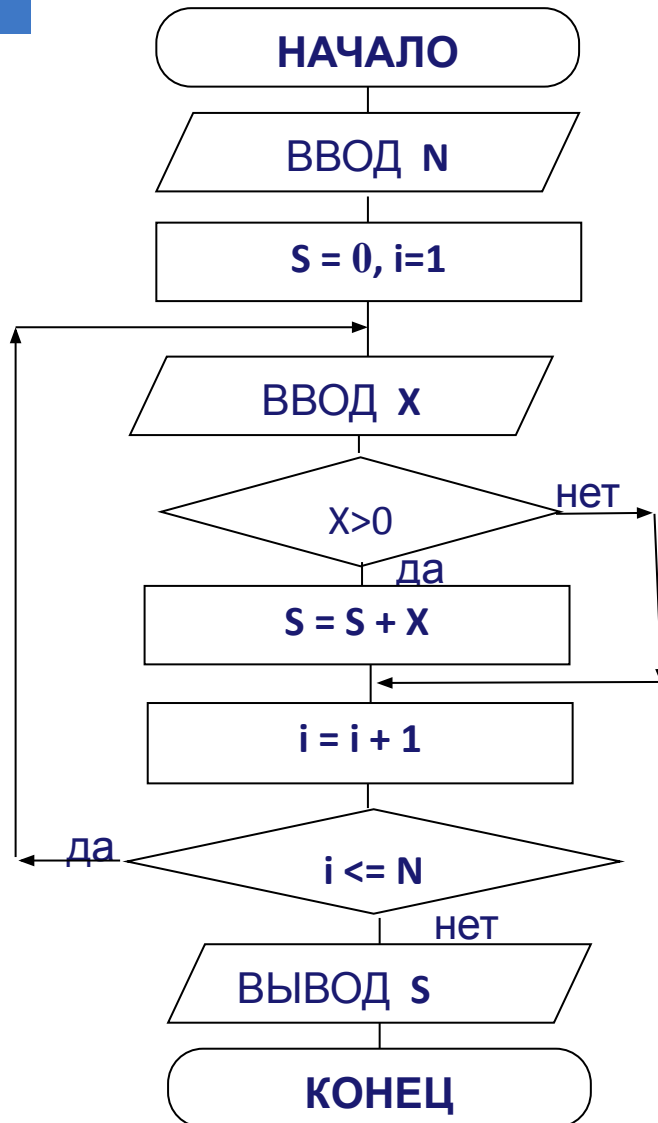
# Цикл с постусловием



- ❖ В цикле с постусловием сначала выполняется **тело** цикла, затем управление передается на **проверку условия**.
- ❖ В зависимости от истинности или ложности условия, тело цикла выполняется повторно или же происходит переход к оператору, следующему за телом цикла.

Цикл с постусловием гарантированно **выполняется хотя бы раз**.

# Цикл с постусловием

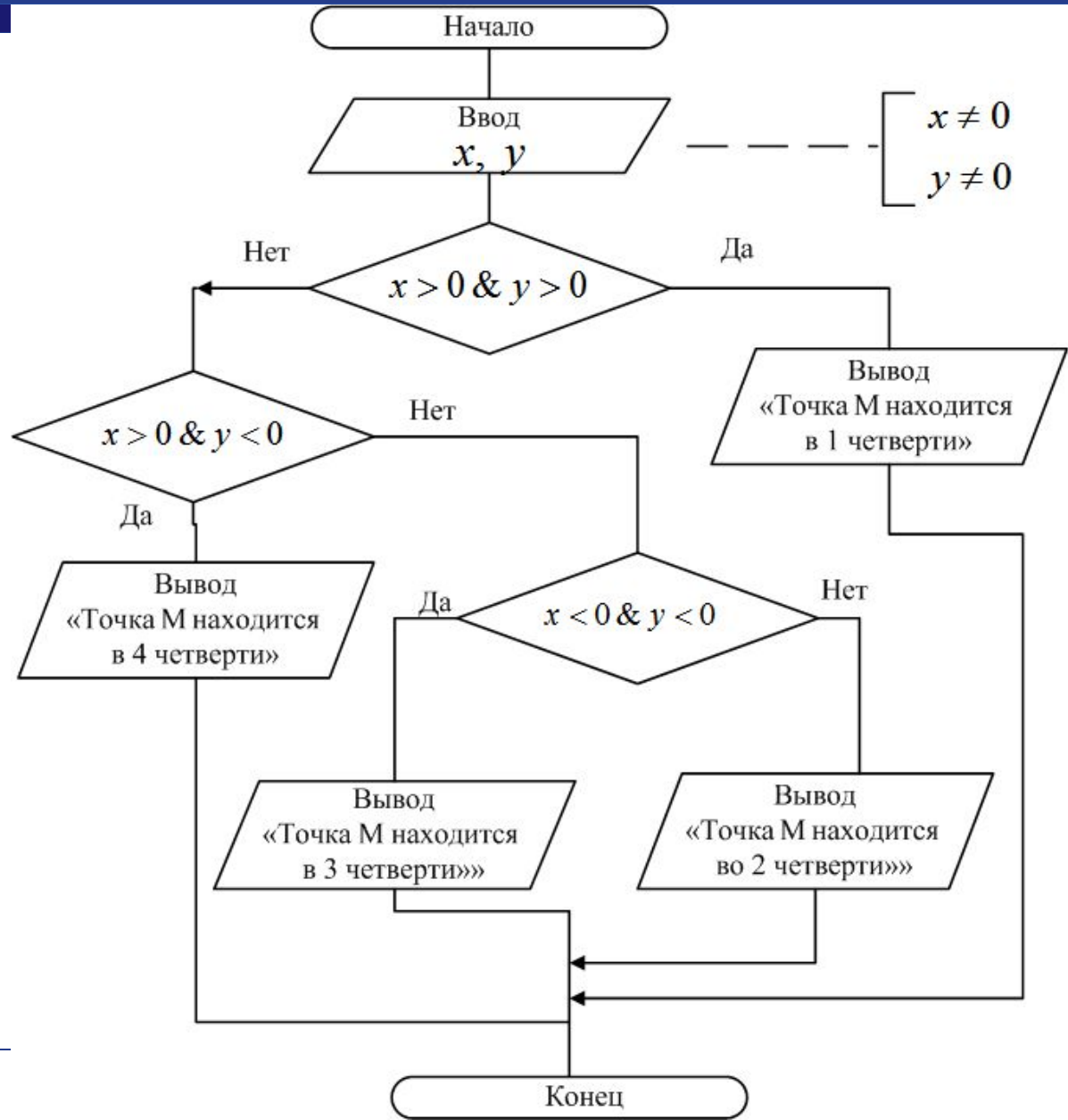


С клавиатуры вводится последовательность из  $N$  чисел. Определить сумму положительных элементов этой последовательности

Условие  $i \leq N$  проверяется после выполнения тела цикла. Поэтому тело цикла выполнится хотя бы один раз

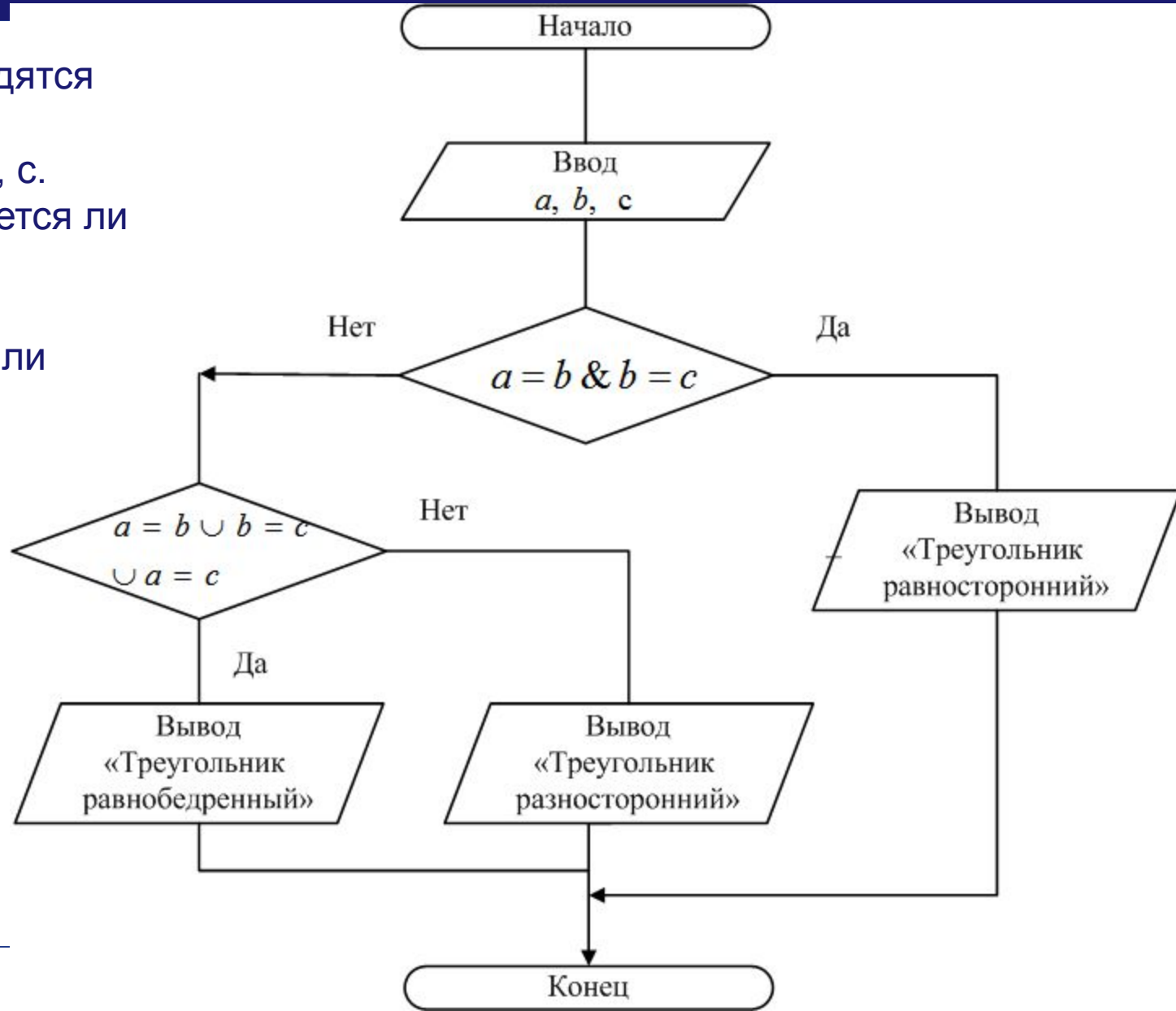
# Примеры

Вводятся ненулевые координаты точки  $M(x,y)$ .  
Определить к какой четверти координатной плоскости принадлежит точка  $M$

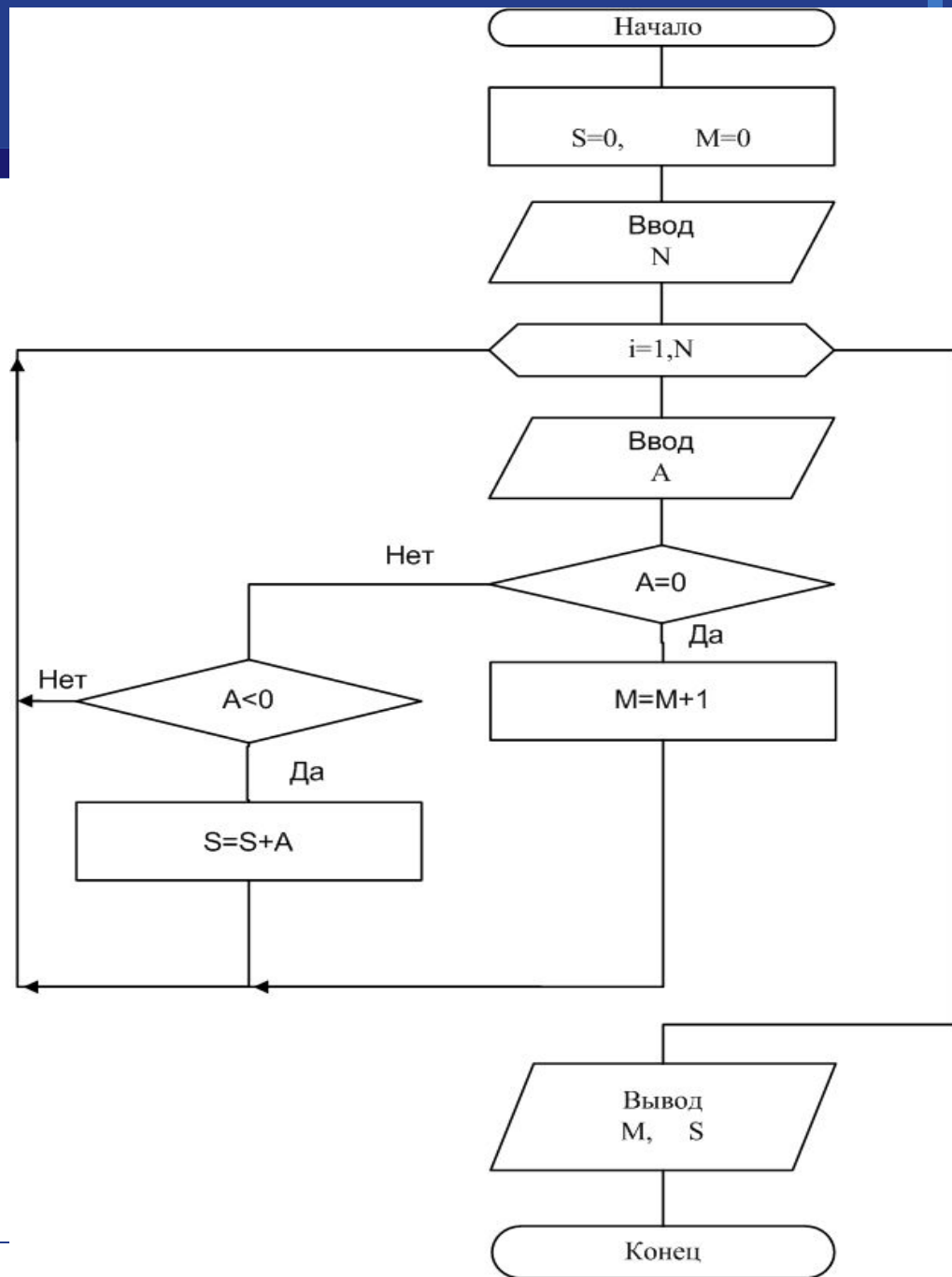


# Примеры

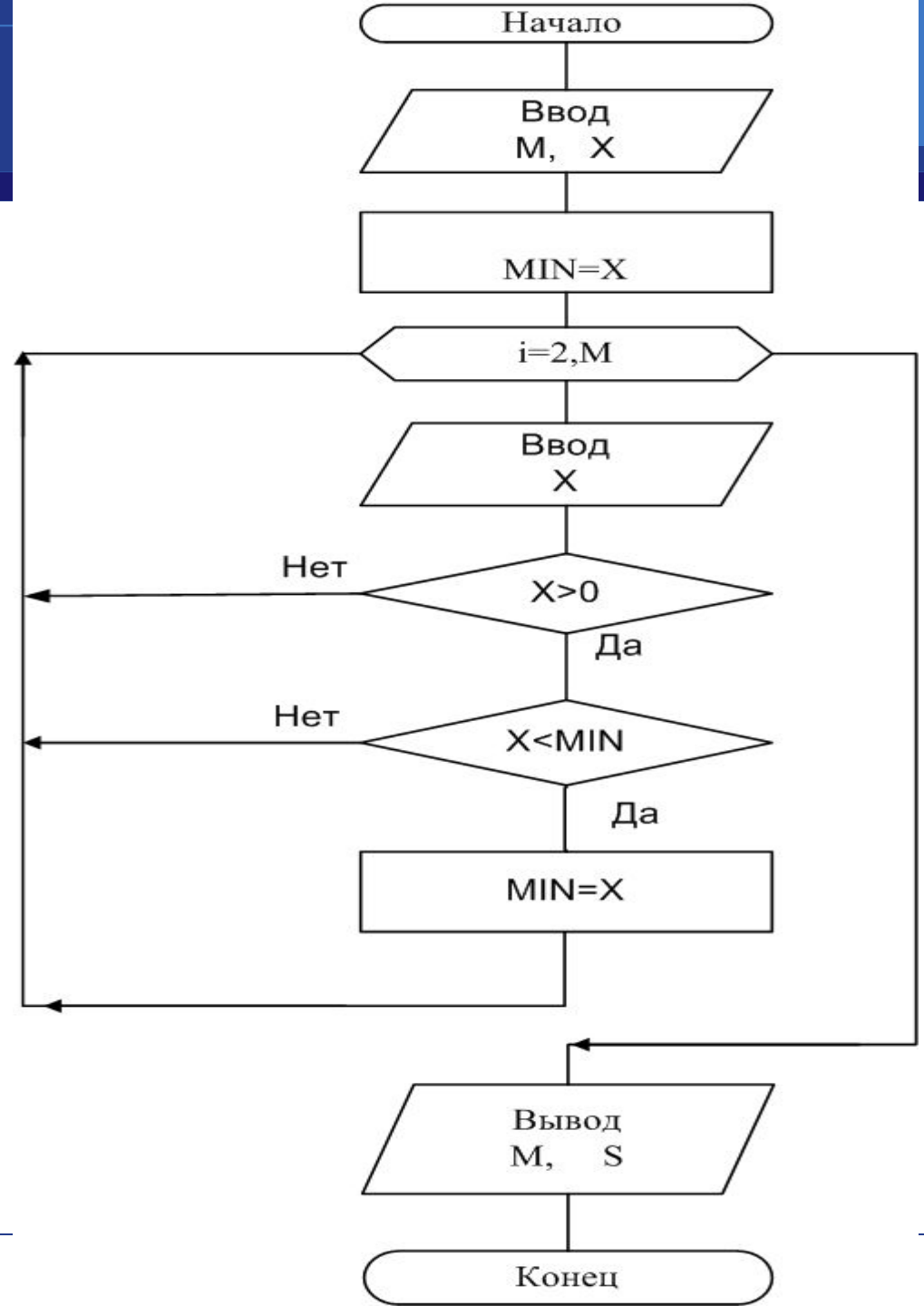
С клавиатуры вводятся  
размеры сторон  
треугольника:  $a, b, c$ .  
Определить, является ли  
треугольник  
равнобедренным,  
равносторонним или  
разносторонним



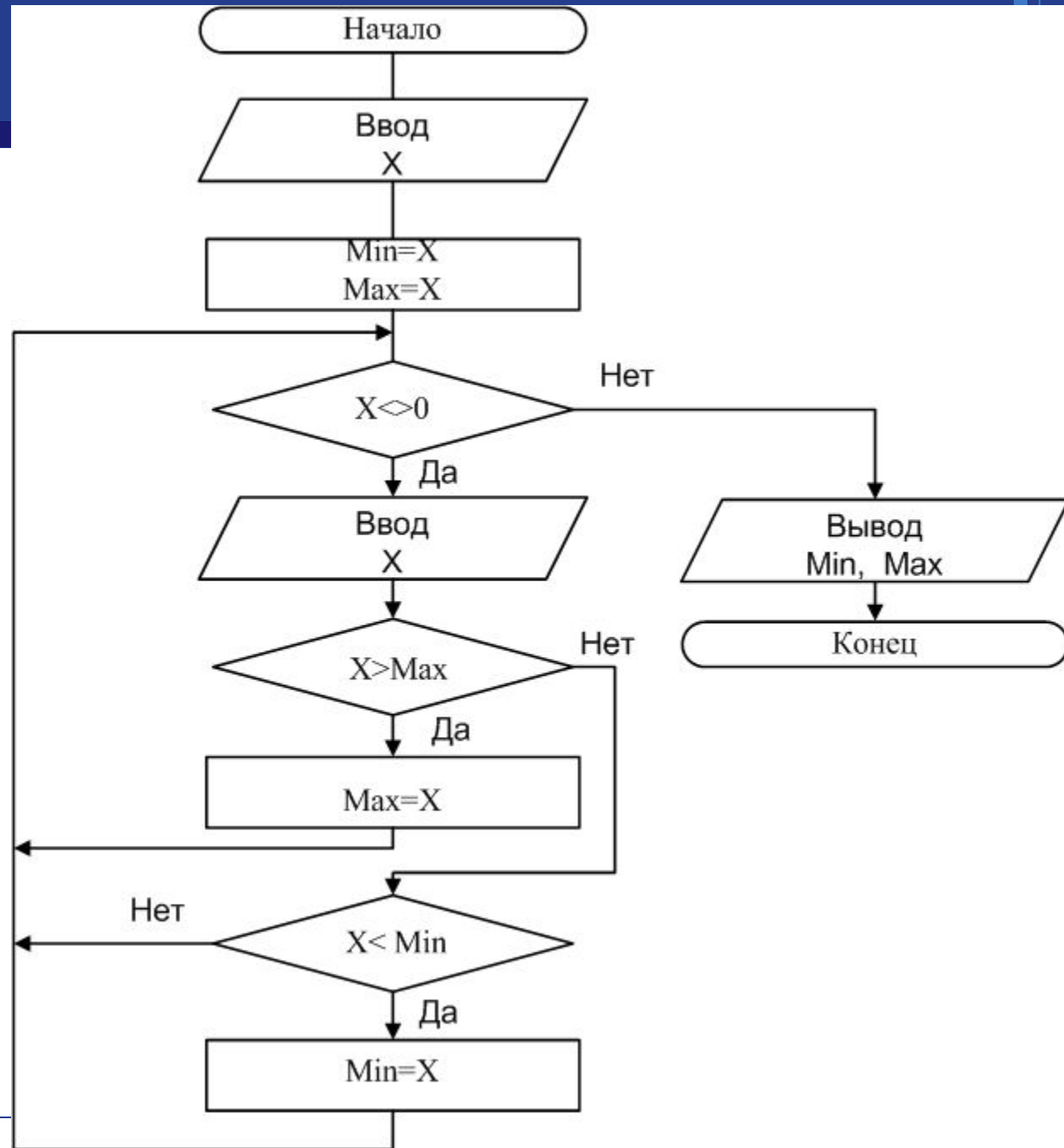
С клавиатуры вводится последовательность из  $N$  чисел. Определить количество нулей и сумму отрицательных элементов этой последовательности



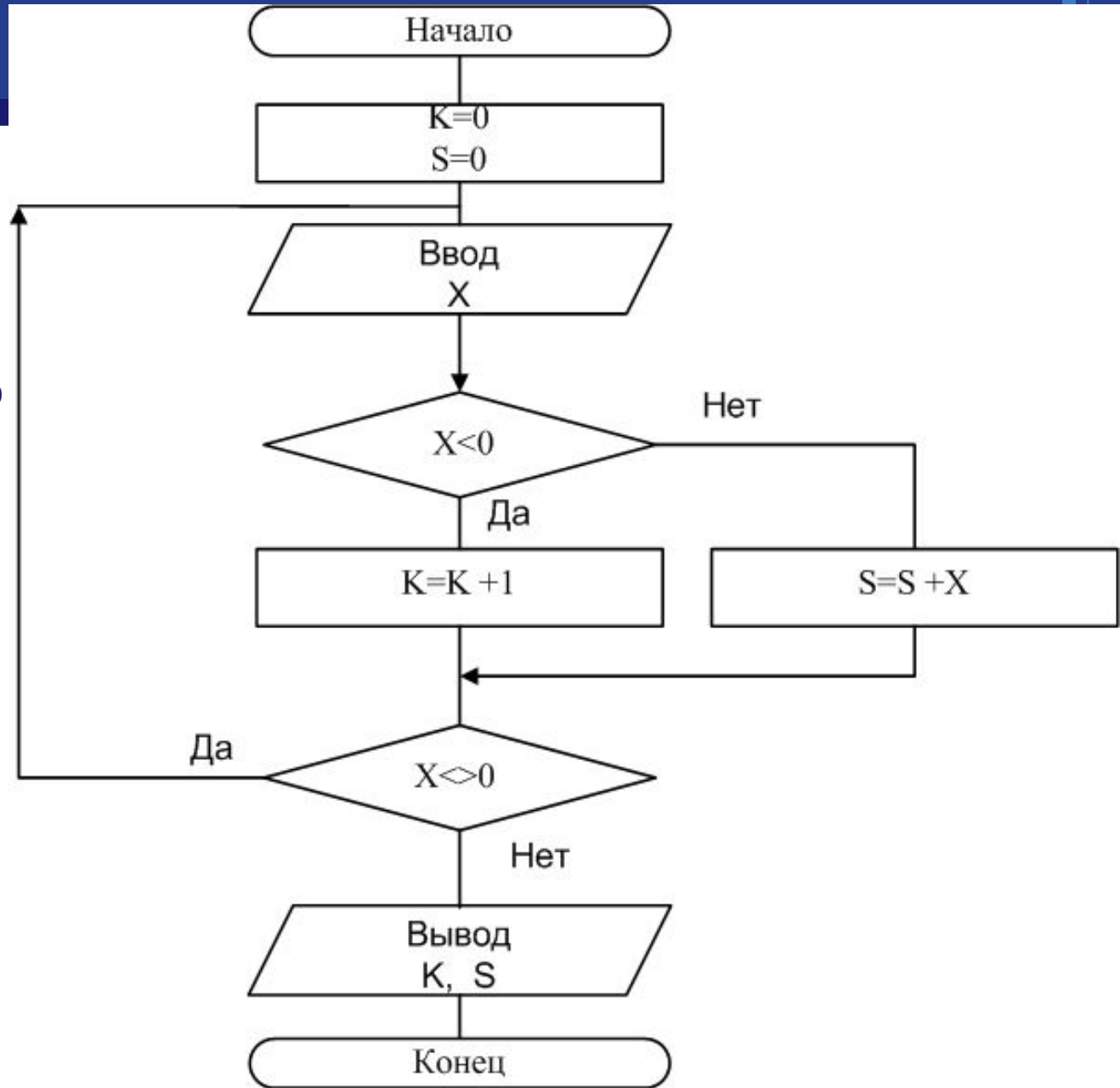
С клавиатуры вводится последовательность из  $N$  чисел. Определить минимальный положительный элемент этой последовательности



С клавиатуры вводится последовательность чисел. Ноль – конец последовательности. Определить минимальный и максимальный элементы этой последовательности



С клавиатуры вводится последовательность чисел. Ноль – конец последовательности. Определить количество отрицательных и сумму положительных элементов этой последовательности





# Базовые алгоритмические конструкции

Различают *алгоритм линейной, разветвляющейся и циклической структуры*, а также *алгоритмы со структурой вложенных циклов*.

*Алгоритмы решения сложных задач могут включать все перечисленные структуры*, которые используются для реализации определенных участков общего алгоритма

---