

# Основы алгоритмизации. Базовые алгоритмические конструкции

Лекция



# Вопросы лекции

1. Понятие, свойства и способы представления алгоритма.
  2. Алгоритмы линейной структуры.
  3. Алгоритмы разветвляющейся структуры.
  4. Алгоритмы циклической структуры.
-

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.



# Происхождение термина «алгоритм»

Слово алгоритм произошло от имени среднеазиатского ученого Аль-Хорезми. В 1857г. в библиотеке Кембриджского университета был найден перевод на латинский язык математической работы Аль-Хорезми, в котором имя Аль-Хорезми упоминается как Алгоритми, откуда и появилось слово «алгоритм».



Мухаммед ал-Хорезми  
(ок. 783—ок. 850 гг.)



*В книге «Об индийском счете» Аль-Хорезми сформулировал правила записи натуральных чисел с помощью арабских цифр и правила действий над ними столбиком.*

Понятие алгоритма, являющееся фундаментальным в математике и информатике, возникло задолго до появления средств вычислительной техники.

Первоначально под алгоритмом понимали способ выполнения арифметических действий над десятичными числами.

В дальнейшем это понятие стали использовать для обозначения любой последовательности действий, приводящей к решению поставленной задачи.



# Термин «алгоритм» в бытовом понимании

В повседневной жизни выполнение каждой, даже простой задачи обычно осуществляется в несколько последовательных этапов (шагов).

Подобную инструкцию – четкую последовательность шагов в решении какой-либо жизненной задачи – принято называть алгоритмом.

Каждое отдельное действие – это шаг алгоритма.



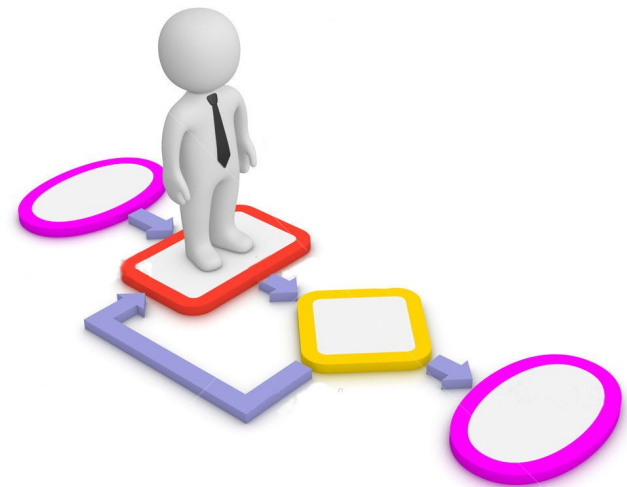
# Пример алгоритма



# Понятие алгоритма

**Алгоритм** – описанная на некотором языке точная конечная система правил, определяющая содержание и порядок действий над некоторыми объектами, строгое выполнение которых дает решение поставленной задачи.

Любой алгоритм предназначен для определенного **исполнителя**.

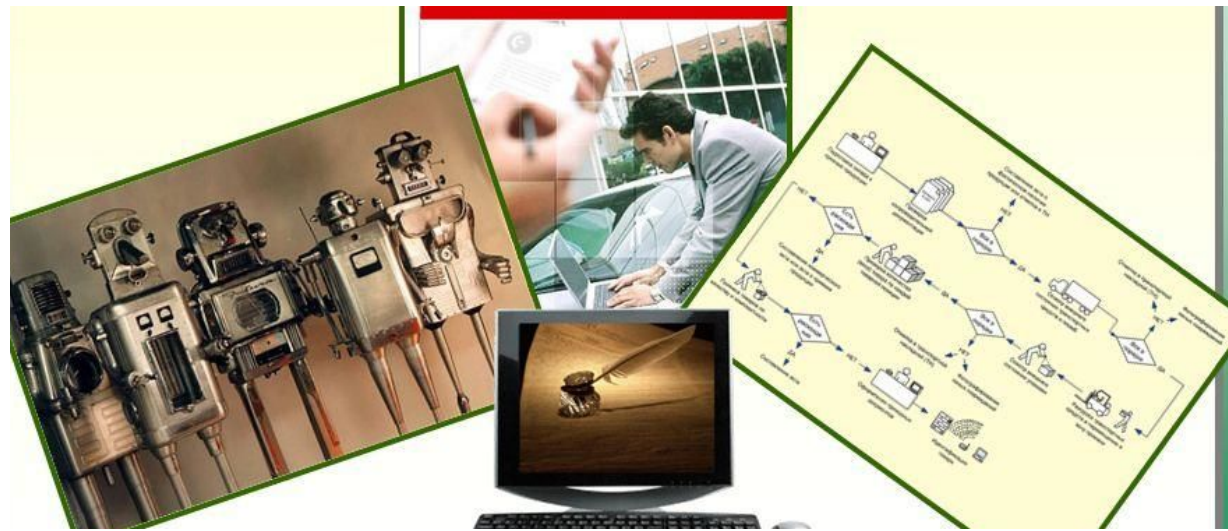




**Исполнитель алгоритма** — это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.

**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**.

# Исполнитель

Формальный

Неформальный

Круг решаемых задач

Среда исполнителя

Область, обстановка, условия

СКИ

Непосредственное управление

Режимы работы

Программное управление

Исполнители алгоритмов



# Понятие алгоритма

**Алгоритм** – точное предписание, состоящее из последовательности действий для некоторого исполнителя, ведущих к решению задачи за конечное число шагов.

Алгоритм моделирует решение задачи в виде точно определенной последовательности действий для некоторого исполнителя по преобразованию исходных данных в результат. Процесс составления алгоритмов называют **алгоритмизацией**.

В информатике универсальным исполнителем алгоритмов является компьютер.

Кафедра информатики

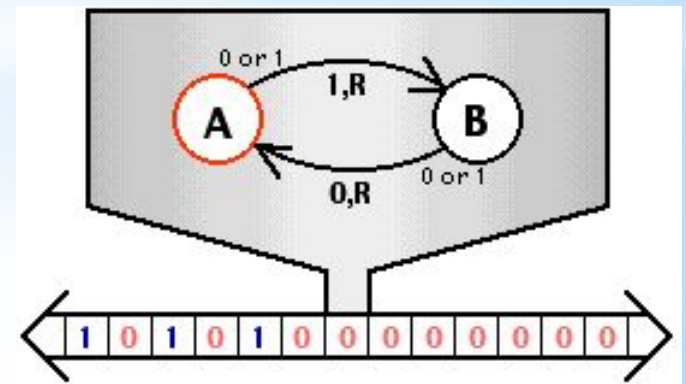


Задача **нахождения единообразной формы записи алгоритмов**, решающих различные задачи, является одной из важнейших в теории алгоритмов.

Предполагается, что каждый шаг алгоритма должен быть таков, что его может выполнить достаточно простое устройство.

Для уточнения понятия «алгоритм» и математического исследования алгоритмов в 30-х гг. XX века были предложены абстрактные вычислительные машины – **машина Поста и машина Тьюринга**.

Было доказано, что если для решения задачи можно построить машину Поста-Тьюринга, то такая задача алгоритмически разрешима.



# Понятие алгоритма

**Формализация задачи** – это процесс выявления характеристик объекта, подлежащих моделированию, и описание их в некоторой форме. Результатом формализации является информационная модель.

На этапе формализации выделяются известные параметры (исходные данные) и параметры, которые следует найти (результаты), а также сформулировать правила получения результата.

Для решения задачи необходимо знать, что дано, что следует получить, какие действия, и в каком порядке следует для этого выполнить. Предписание, определяющее порядок выполнения действий над данными с целью получения искомых результатов, и есть алгоритм.

# Свойства алгоритмов

<b>Дискретность (прерывность)</b>	Алгоритм решения задачи представляется как последовательное выполнение простых шагов.
<b>Определенность (однозначность)</b>	Каждое действие алгоритма должно быть четким и однозначным.
<b>Результативность (конечность)</b>	Алгоритм должен приводить к решению задачи за конечное число шагов или после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения.
<b>Массовость</b>	Алгоритм решения задачи может применяться к некоторому классу задач, различающихся лишь исходными данными.

# Способы представления алгоритмов

<b>Словесный</b>	Самый простой способ представления алгоритма, который содержит описание алгоритма на естественном языке, например, на русском. Разновидность – формульно-словесный.
<b>Графический (блок-схема)</b>	Алгоритм изображается в виде последовательности связанных между собой блоков, каждый из которых соответствует выполнению одного или нескольких действий. В блок-схеме каждому типу действий соответствует определенная геометрическая фигура (блок). Блоки соединяются линиями переходов, которые определяют очередность выполнения действий.
<b>Программа на алгоритмическом языке</b>	Запись алгоритма на языке программирования, в которой алгоритм представляется в виде последовательности



# СПОСОБЫ ОПИСАНИЯ АЛГОРИТМОВ

**Пример:**

Найти значение следующего выражения:  $Y=2A-(X+6)$

словесно-формульный  
(на естественном языке)

1. Ввести значения  $a$  и  $x$ .
2. Сложить  $x$  и  $6$ .
3. Умножить  $a$  на  $2$ .
4. Вычесть из  $2a$  сумму  $(x+6)$ .
5. Вывести  $y$  как результат вычисления выражения.

графический или  
блок-схемный

начало

ввод  $a, x$

$y_1 = x + 6$

$y_2 = 2 * a$

$y = y_2 - y_1$

вывод  $y$

конец

с использованием  
специальных  
алгоритмических языков

Input  $a, x$

$y_1 = x + 6$


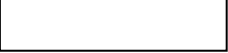
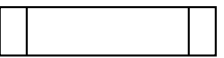

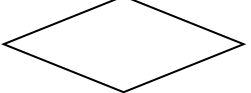

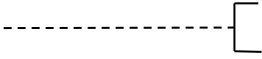

$y_2 = 2 * a$

$y = y_2 - y_1$

Print  $y$

End

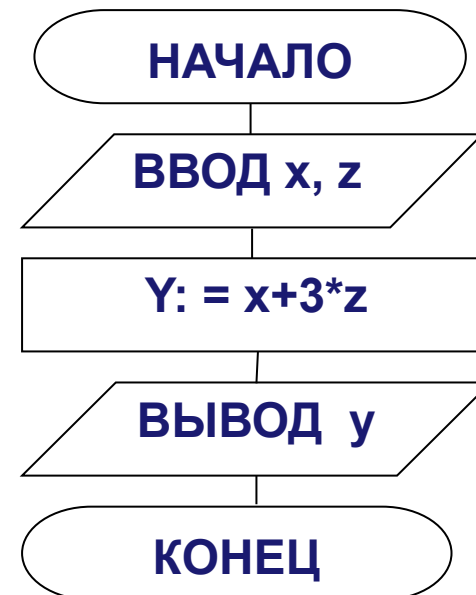
# Графическое представление алгоритма

Наименование	Блок	Функция
Терминатор		Обозначение начала и конца алгоритма.
Процесс		Обработка данных любого вида, выполнение действия.
Данные		Операции ввода и вывода данных.
Предопределенный процесс		Процедура или функция. Одна или несколько операций, которые определены в другом месте
Подготовка		Цикл с заданным числом повторений.
Решение		Условный оператор или переключатель.
Соединитель		Применяется, если блок-схема разбивается на несколько частей.
Комментарий		Используют для пояснительных записей (комментария), в целях объяснения или примечания.
Линия		Линия перехода, поток управления. При переходе влево и вверх добавляются стрелки

# Алгоритмы линейной структуры

**Базовая структура «следование»** (линейная структура) образуется последовательностью действий, следующих одно за другим.

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



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

**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*.

---

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 2 целых значений с отбрасыванием остатка
2	mod	Взятие остатка от деления 2 целых значений
	+	Сложение
	-	Вычитание

❖ 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  $c = 20,$   
 $n = 12$**

# ЭТАПЫ РЕШЕНИЯ ЗАДАЧИ:

Этапы  
решения  
задачи

Постановка задачи

Построение математической модели

Конструирование алгоритма

Перевод алгоритма в программу

Ввод и тестирование программы

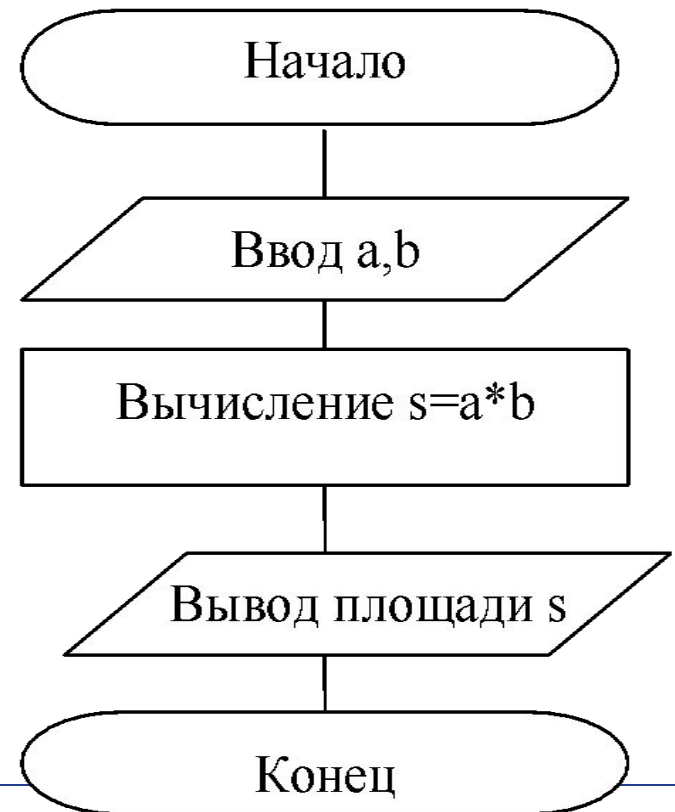
Получение и анализ результатов  
решения задачи



# Пример

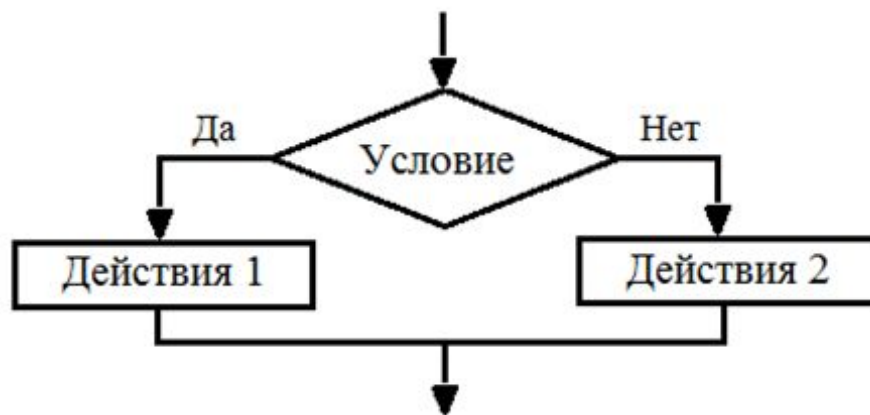
Составить алгоритм вычисления площади прямоугольника  $s$  по известным длинам сторон  $a$ ,  $b$ .

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

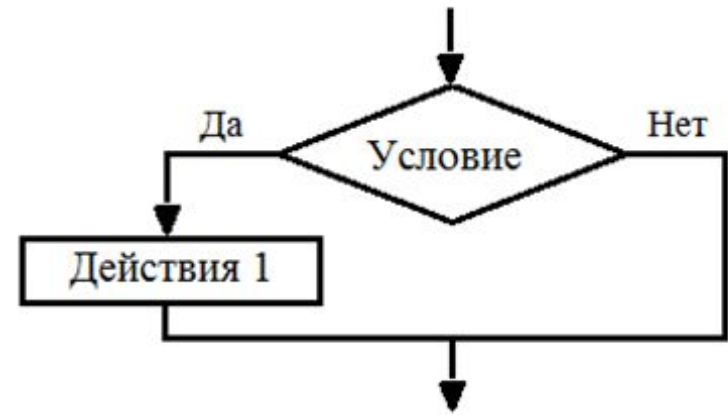


# Алгоритмы разветвляющейся структуры

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



Полная форма

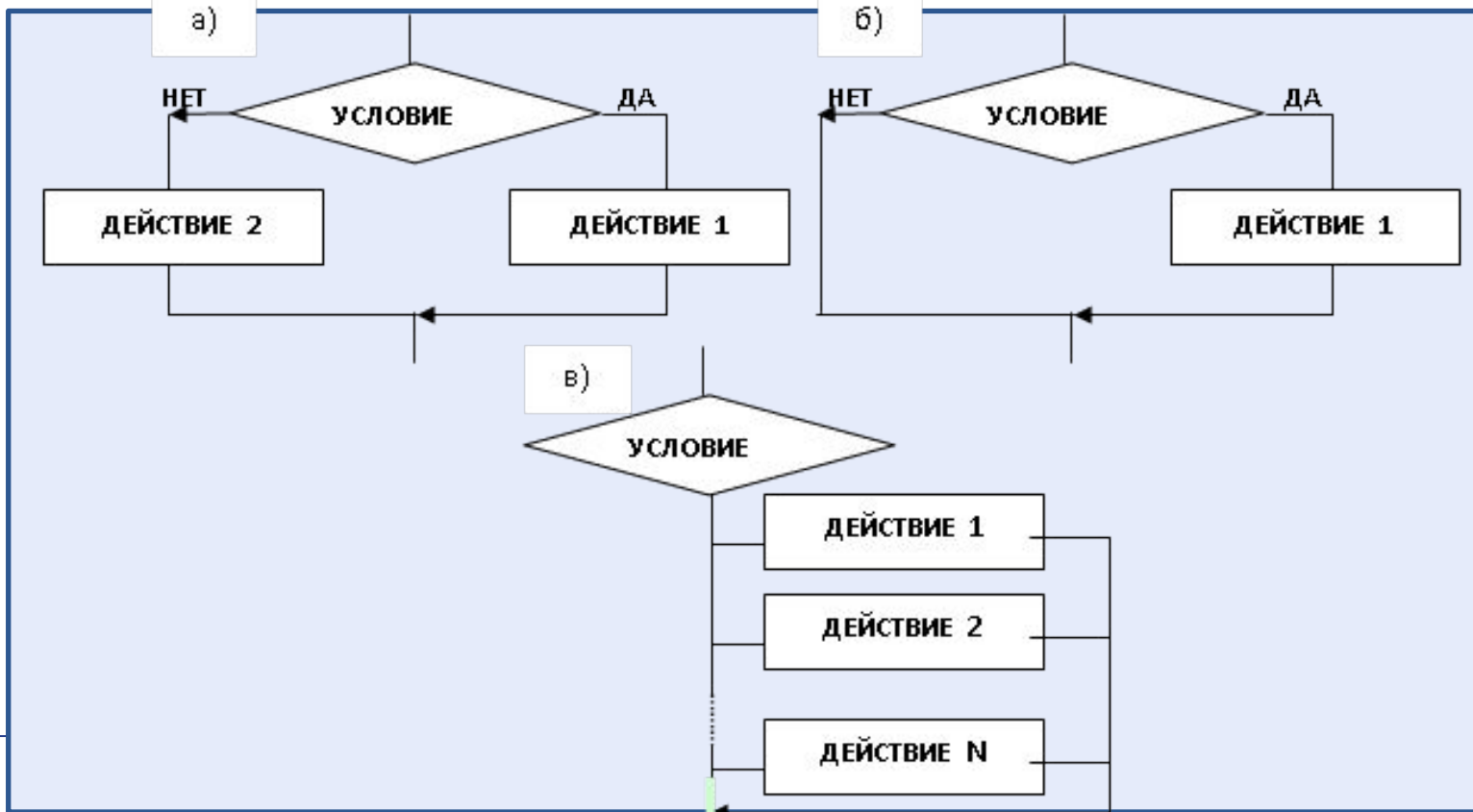


Сокращенная форма

# Алгоритмы разветвляющейся структуры

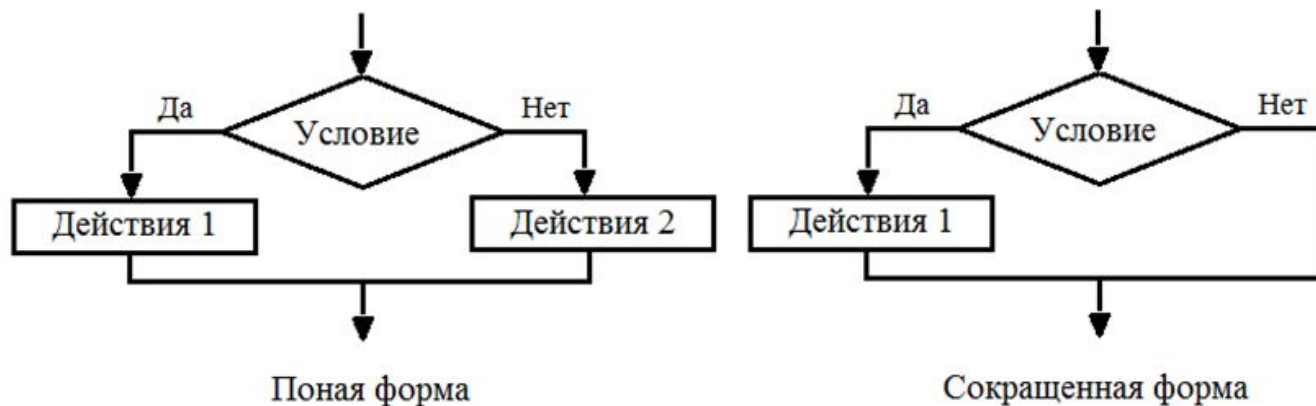
Структура «ветвление» существует в трёх основных вариантах:

**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»,
- $>=$  - «больше или равно»
- $<=$  - «меньше или равно».

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

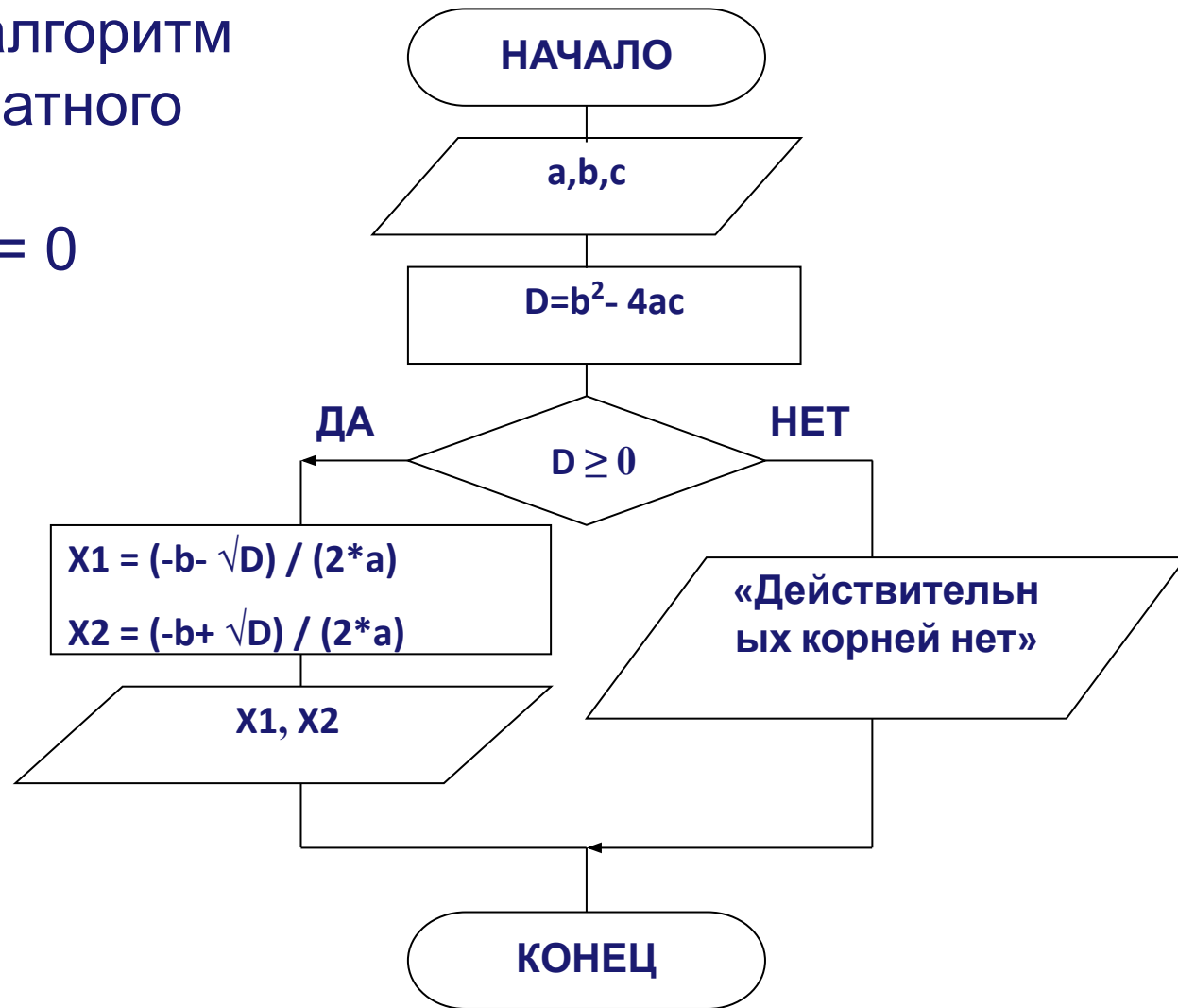
**Составное условие** – это несколько простых условий, соединённых логическими операциями

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

$(A \geq 10) \text{ и } (A \leq 99)$

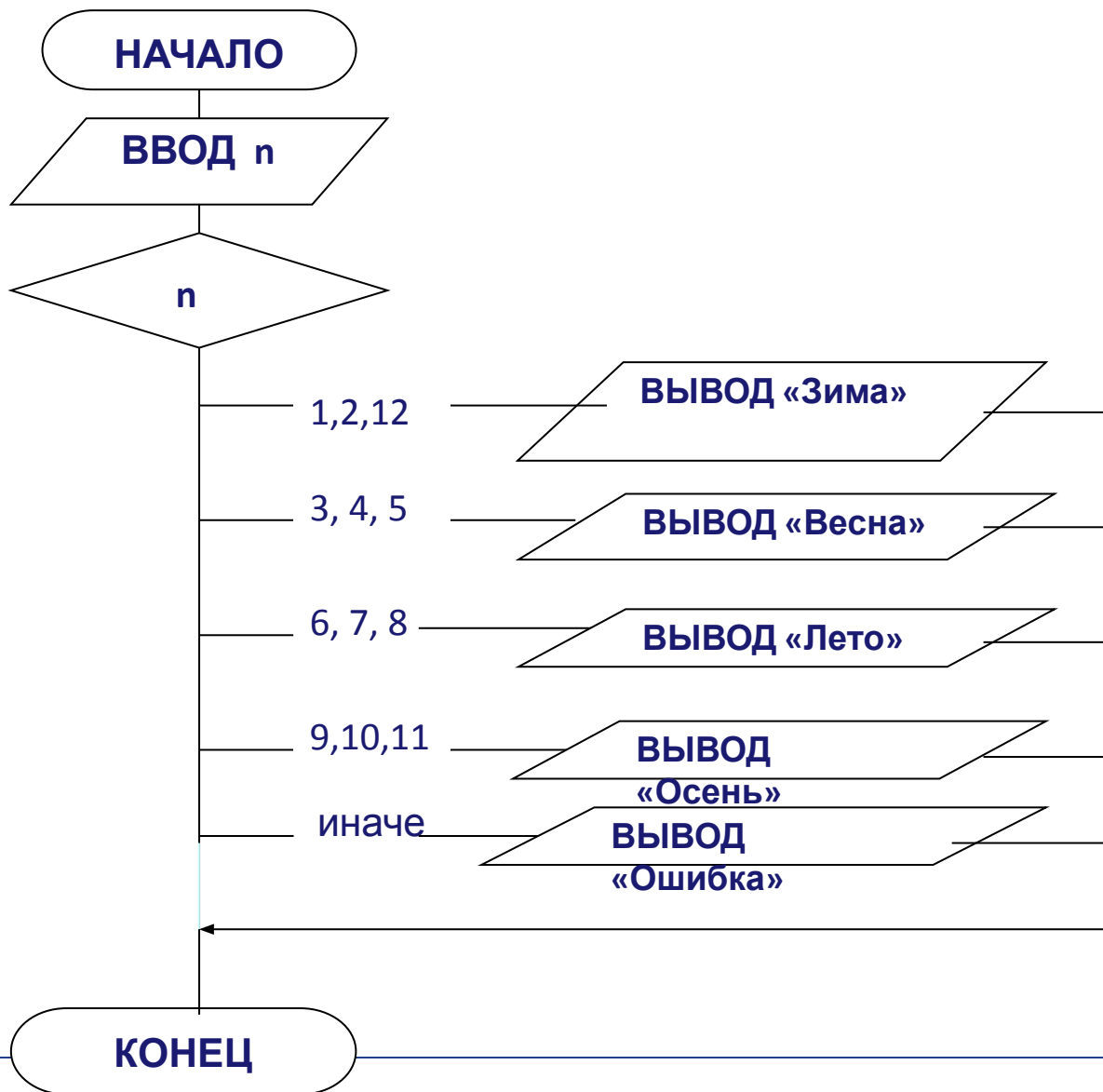
# Пример алгоритма ветвления

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



# Алгоритмы циклической структуры

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

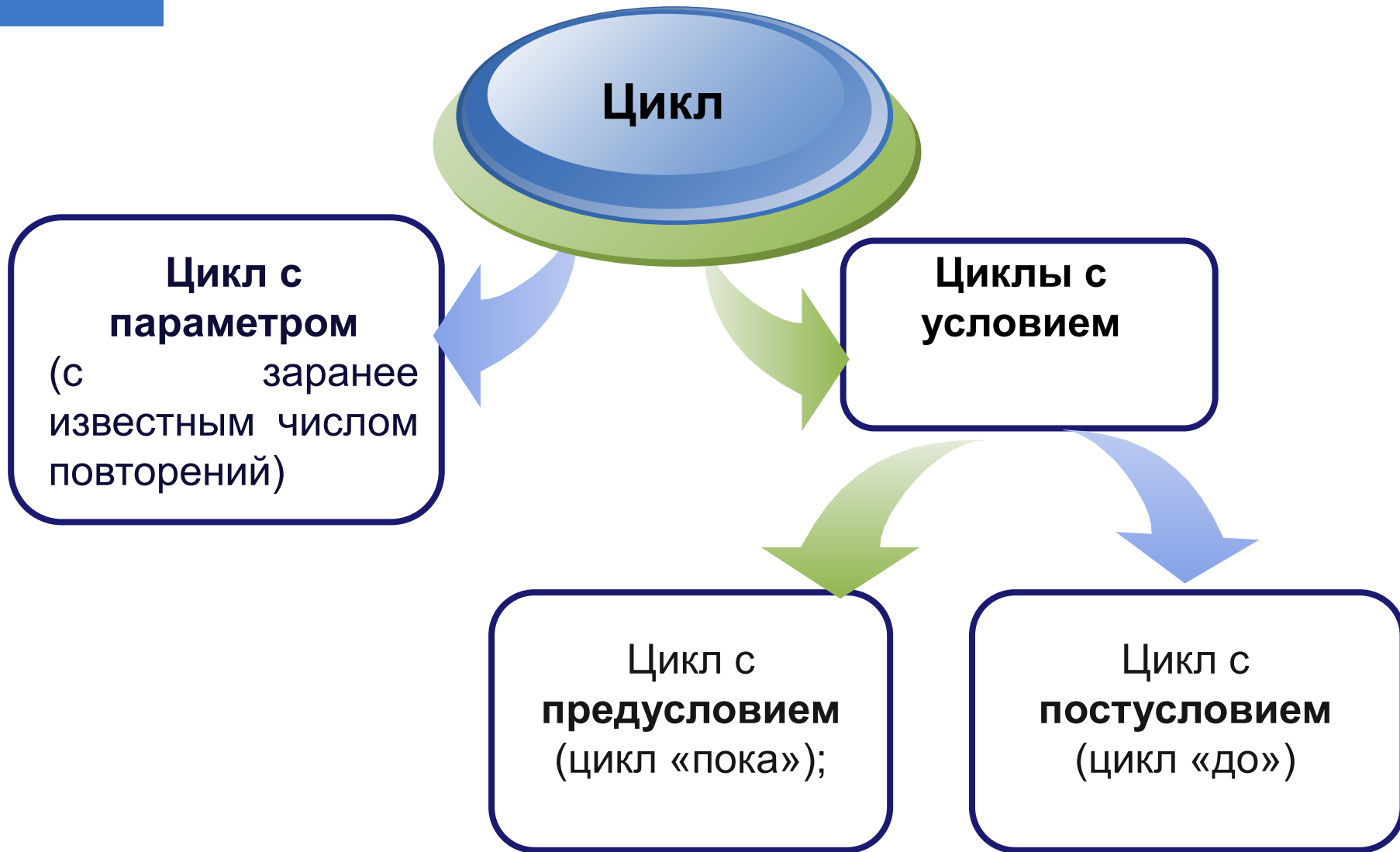
Повторяющаяся совокупность действий называется – **телом цикла**.

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

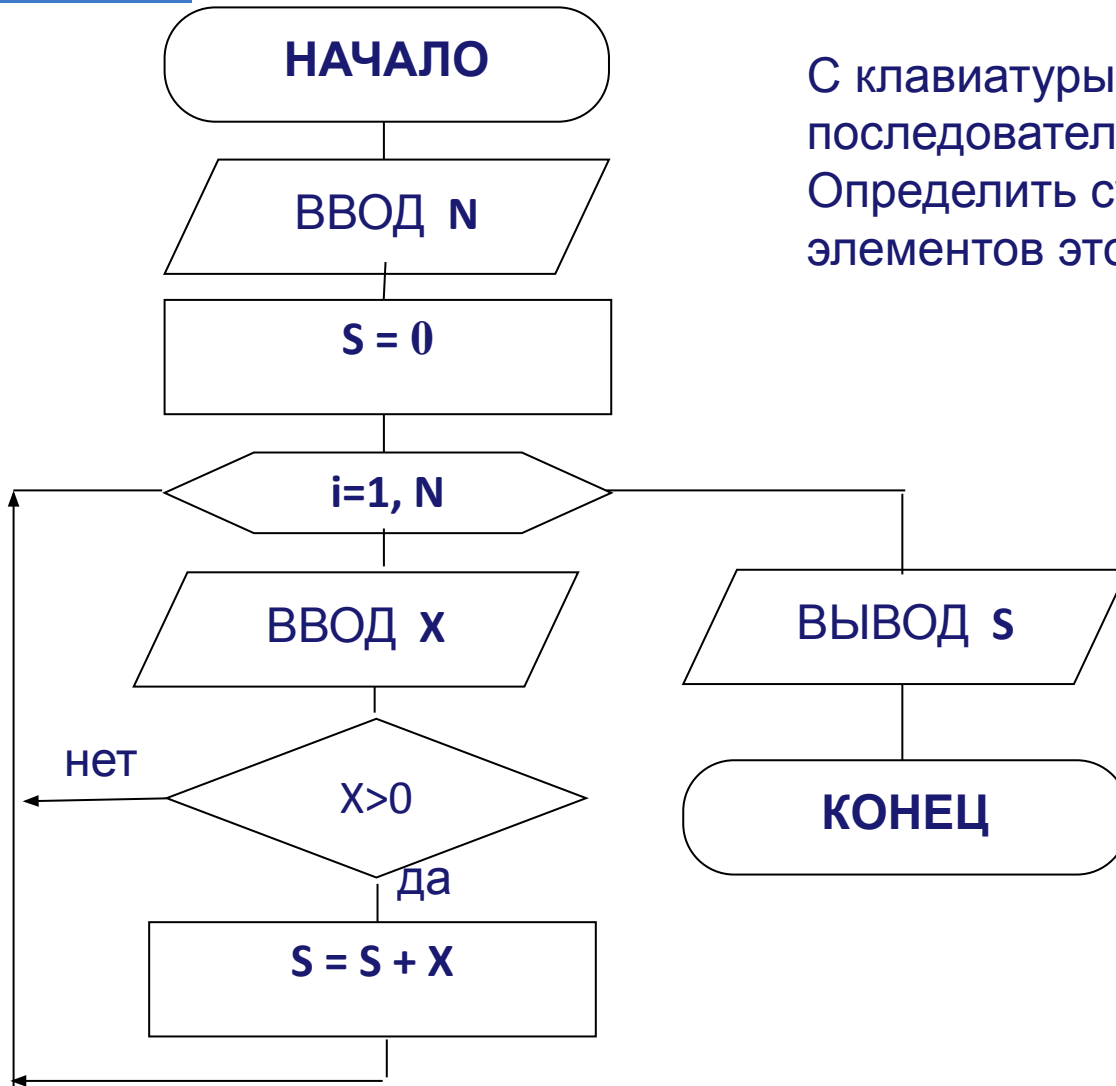


## *Работа цикла*

- 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



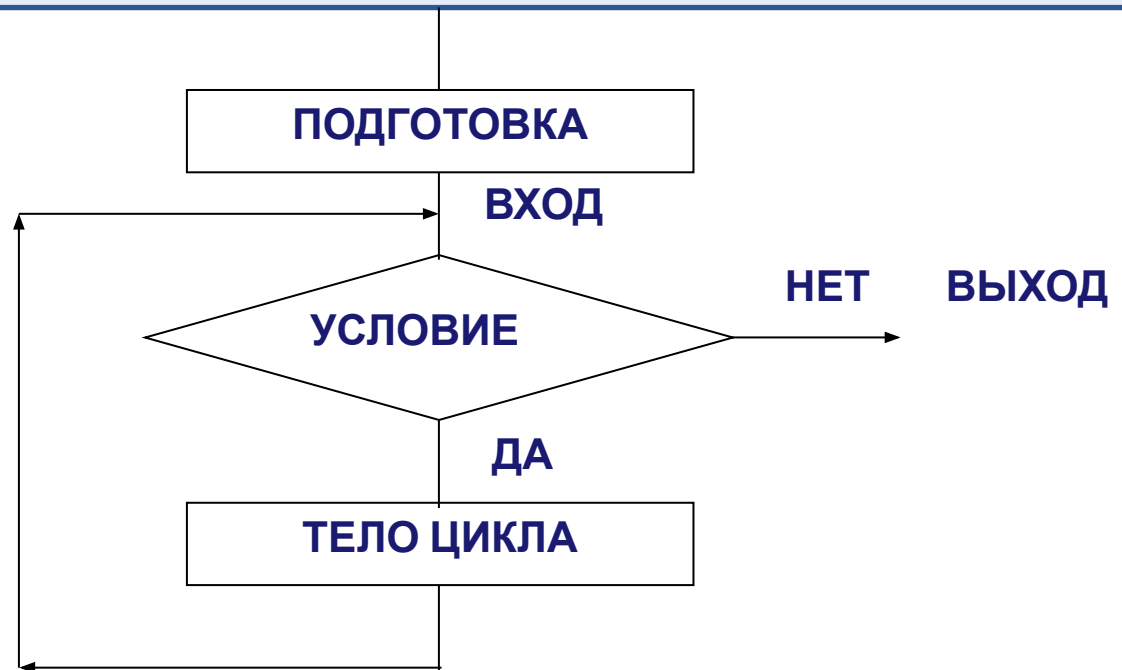
# Цикл с предусловием

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  $h$

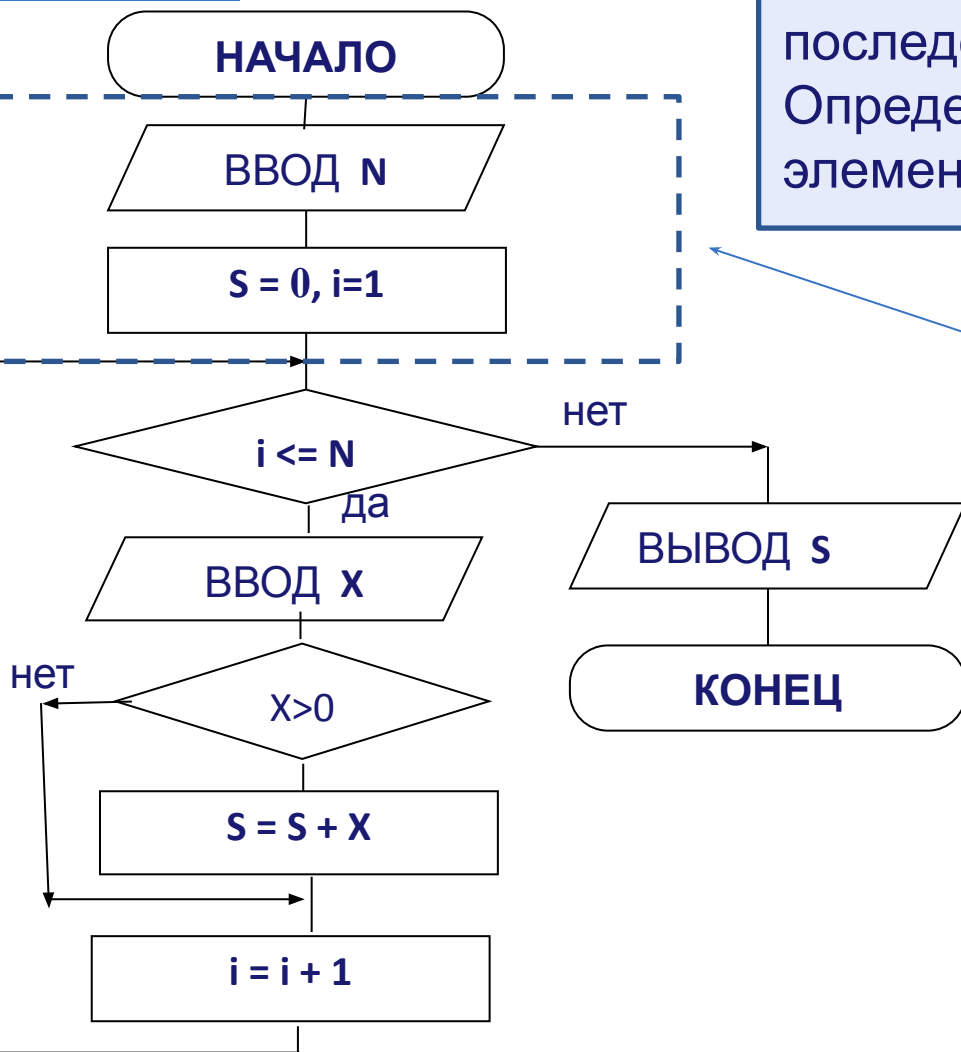
В теле цикла значение параметра цикла увеличивается на величину шага  $h$

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



# Цикл с предусловием

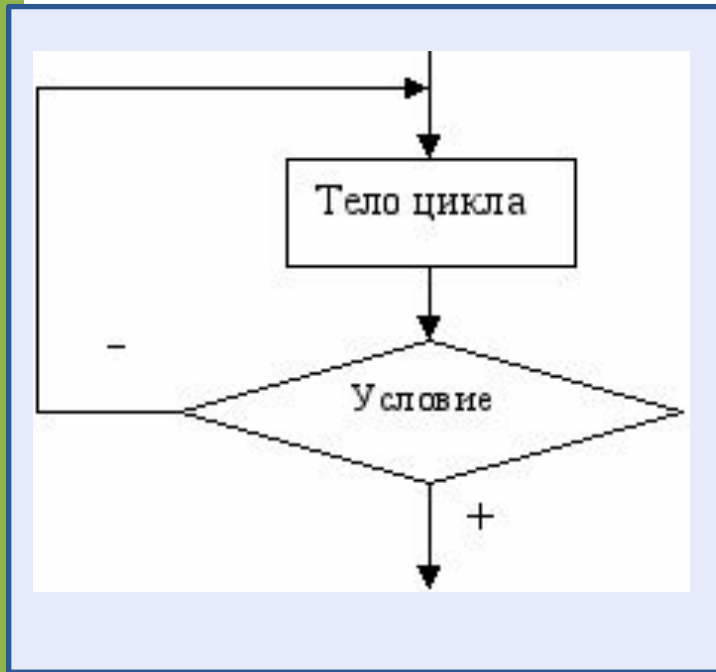
С клавиатуры вводится последовательность из  $N$  чисел. Определить сумму положительных элементов этой последовательности



**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, производятся выход из цикла и вывод результата

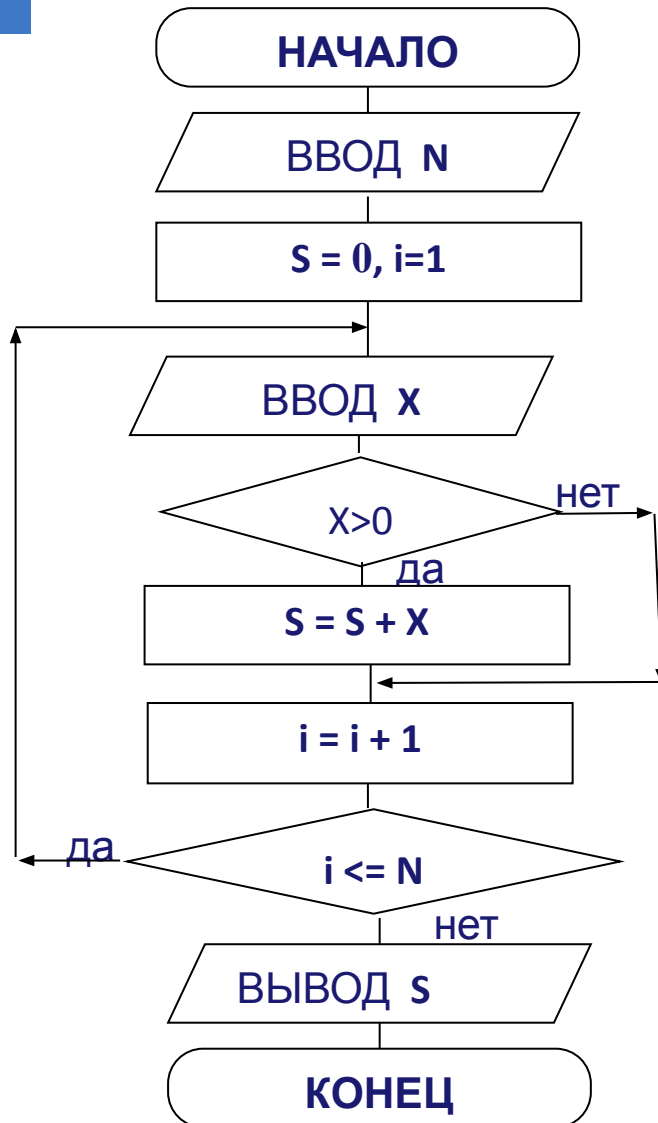
# Цикл с постусловием



- ❖ 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

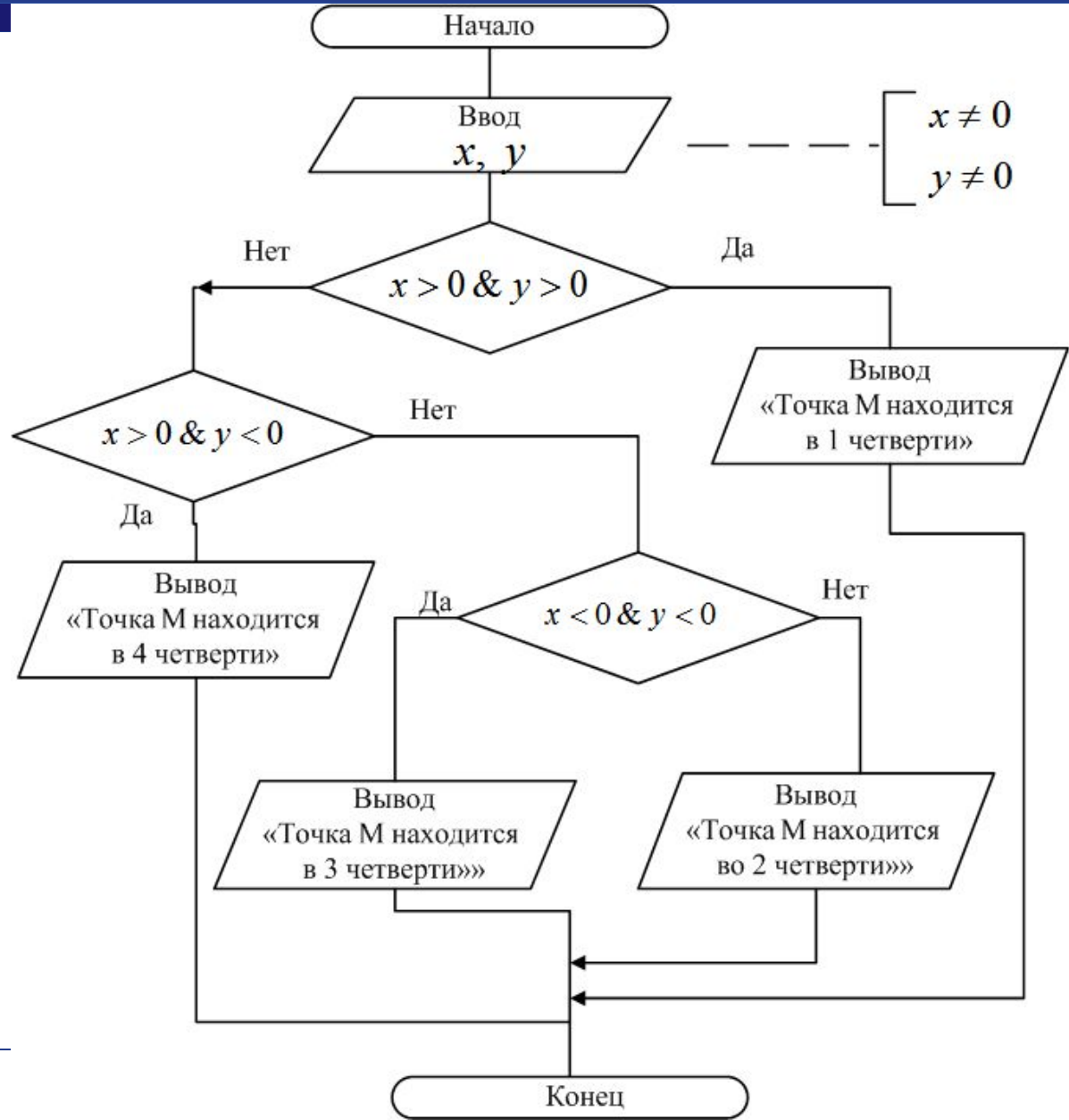


С клавиатуры вводится последовательность из  $N$  чисел. Определить сумму положительных элементов этой последовательности

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

# Примеры

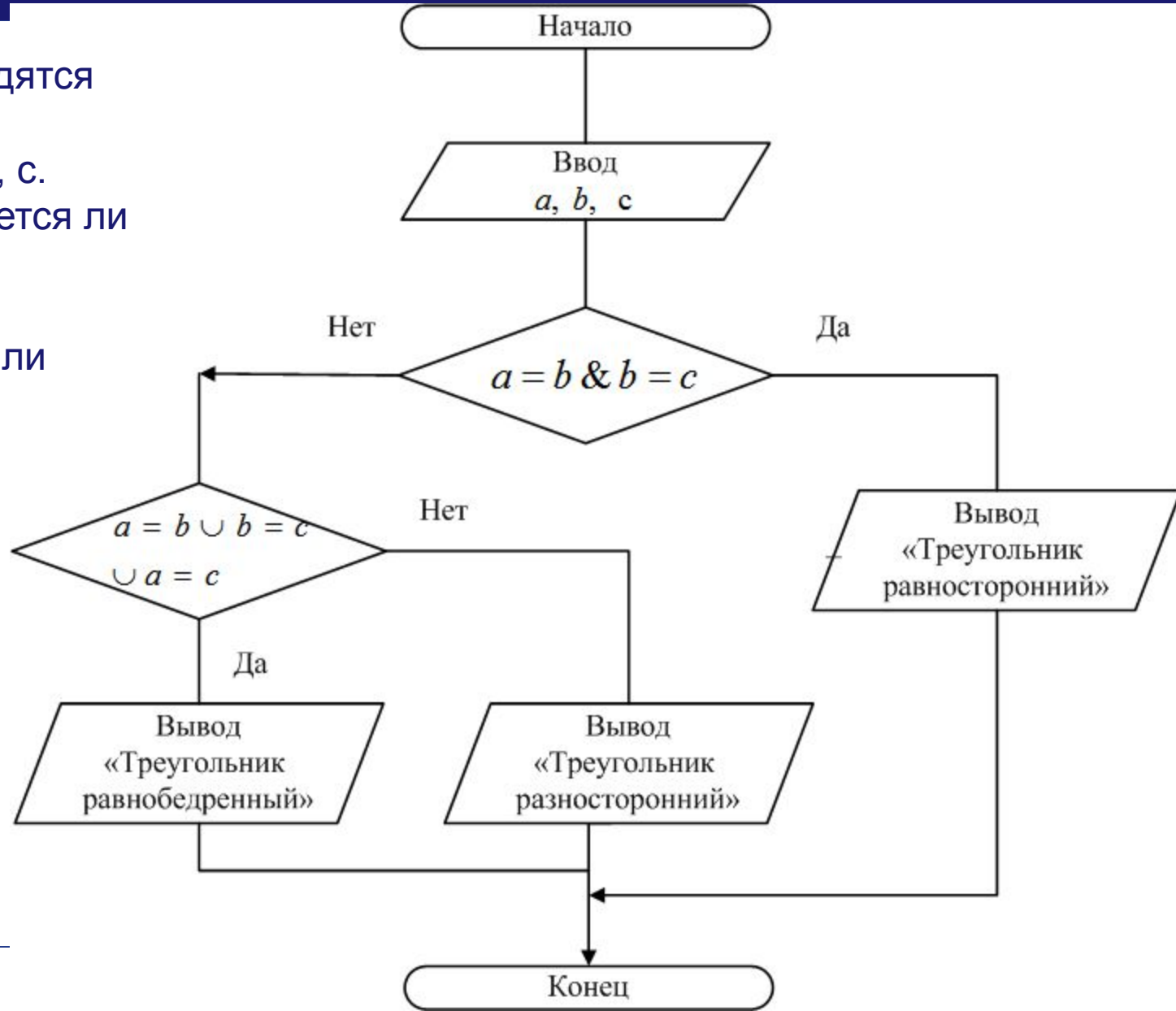
Вводятся ненулевые координаты точки  $M(x,y)$ .  
Определить к какой четверти координатной плоскости принадлежит точка  $M$



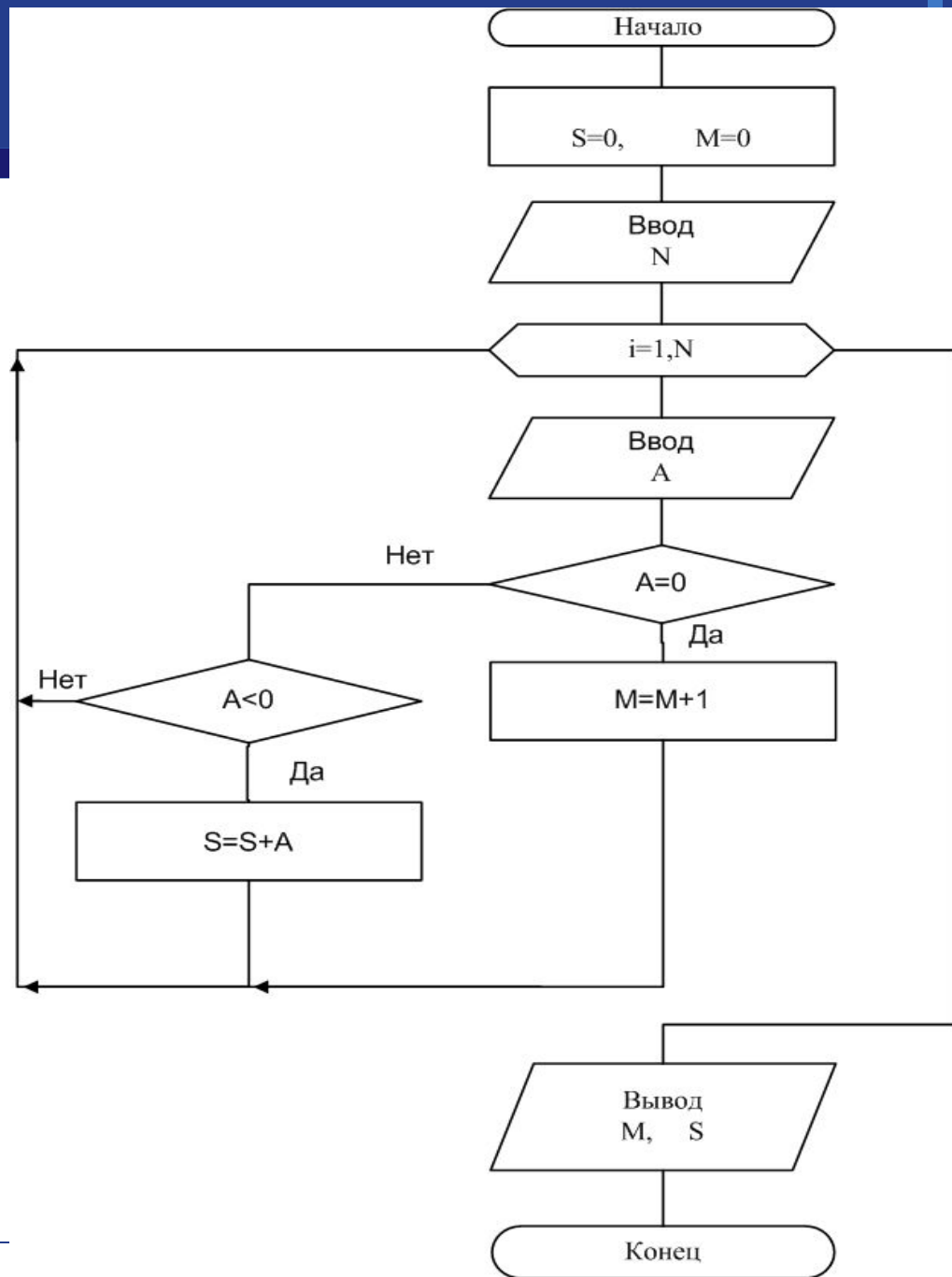


# Примеры

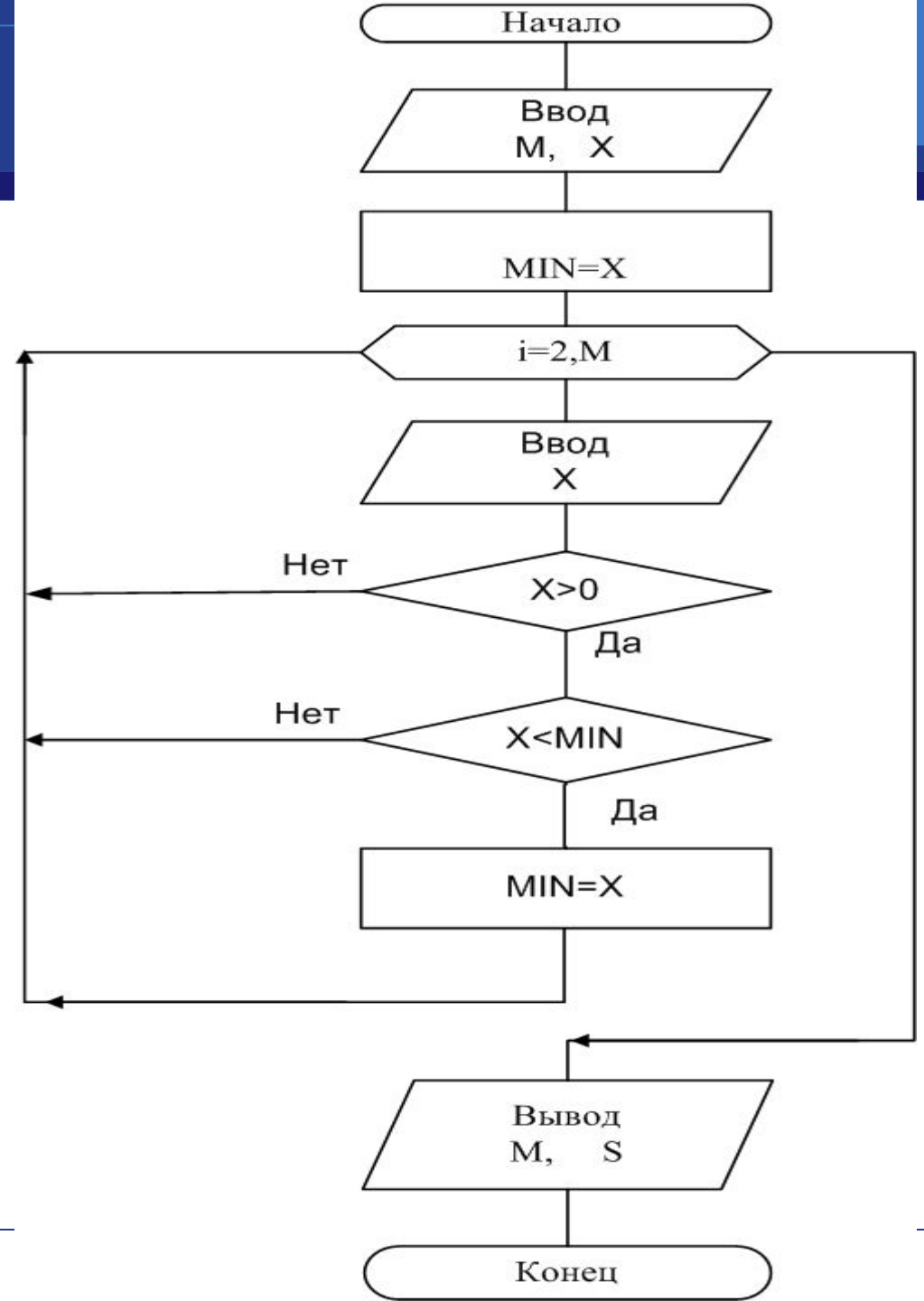
С клавиатуры вводятся  
размеры сторон  
треугольника:  $a, b, c$ .  
Определить, является ли  
треугольник  
равнобедренным,  
равносторонним или  
разносторонним



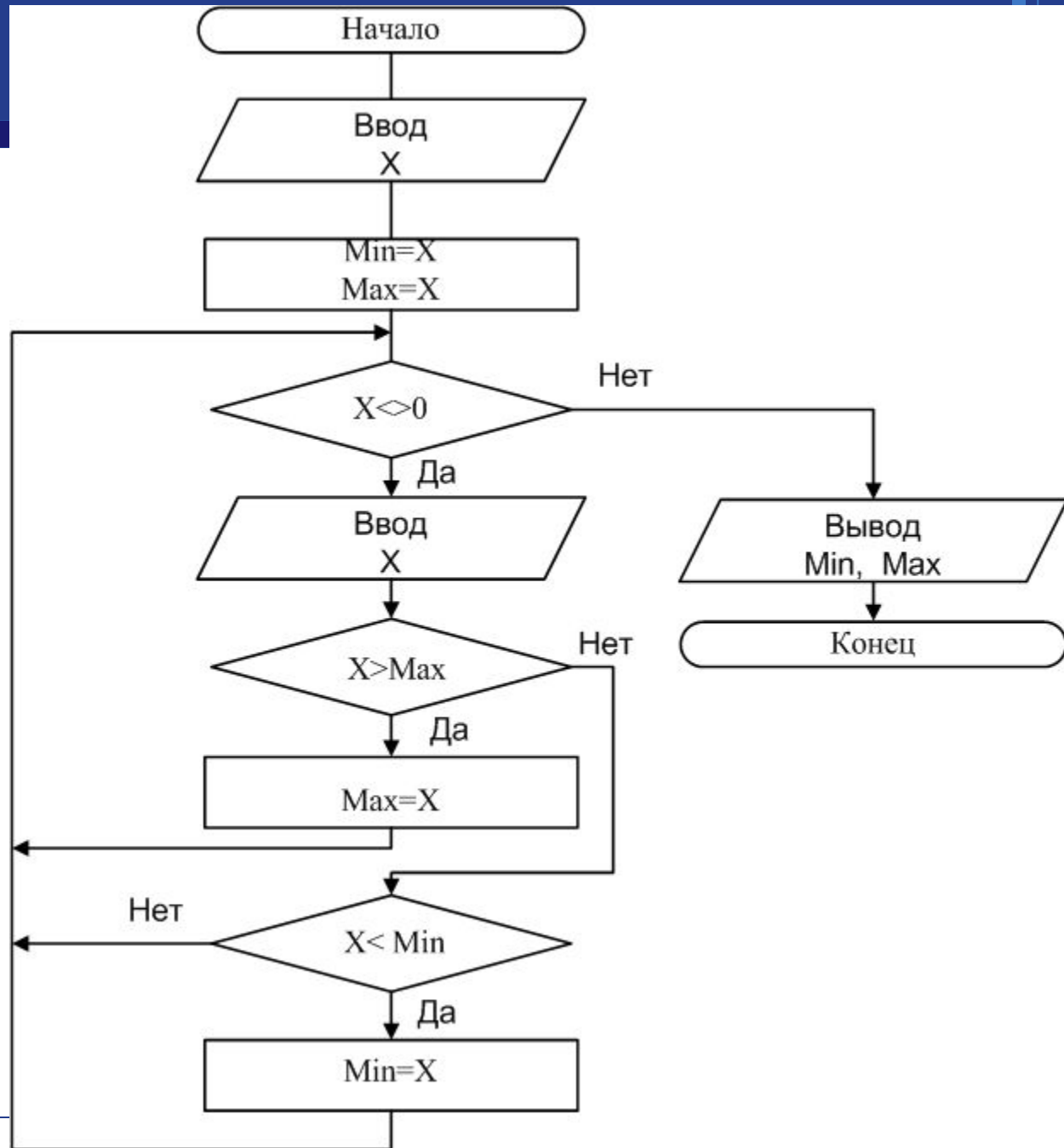
С клавиатуры вводится последовательность из  $N$  чисел. Определить количество нулей и сумму отрицательных элементов этой последовательности



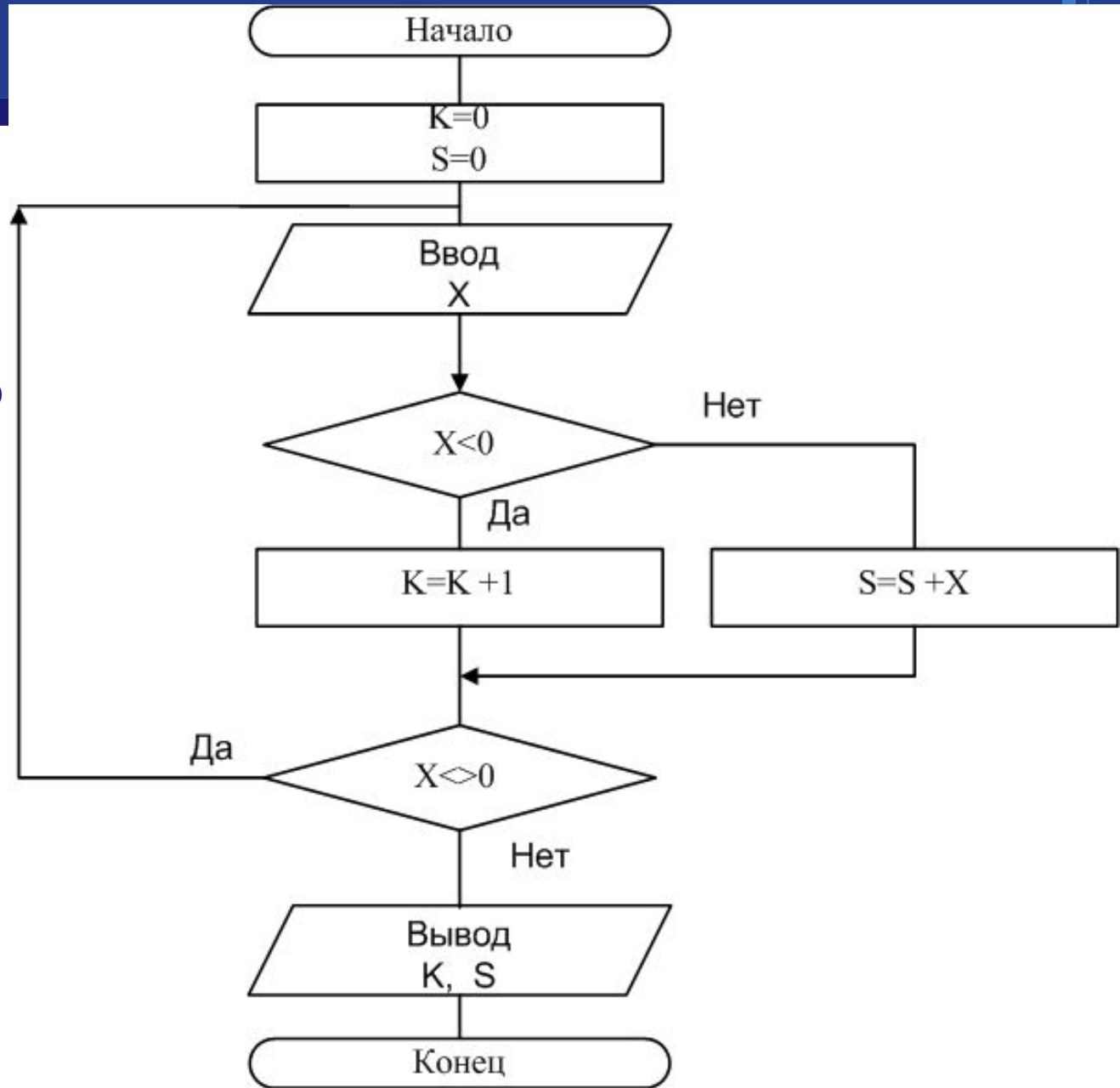
С клавиатуры вводится последовательность из  $N$  чисел. Определить минимальный положительный элемент этой последовательности



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



# Спасибо за внимание!

Кафедра информатики

