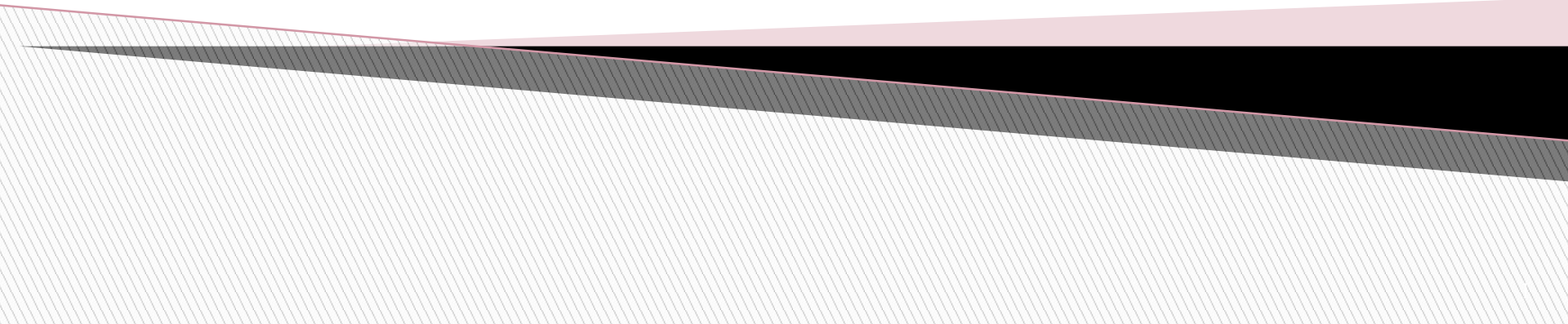


# ОСНОВЫ АЛГОРИТМИЗАЦИИ



**Понятие алгоритма, его свойства.**

**Основные алгоритмические конструкции**

# Понятие алгоритма

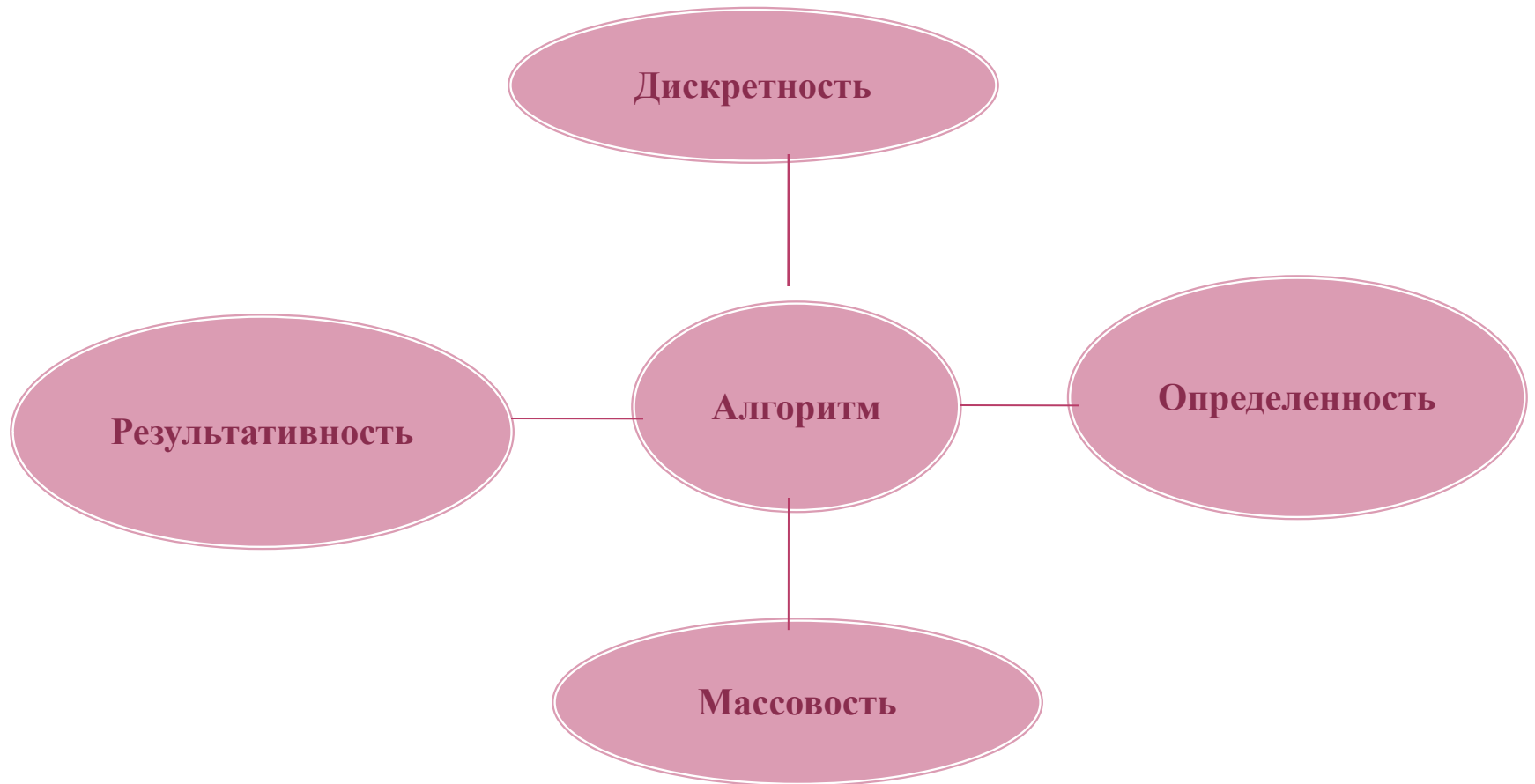
**Алгоритм** –это метод решения задачи, записанный по определенным правилам, обеспечивающим однозначность его понимания и механического исполнения при всех значениях исходных данных.

# Понятие алгоритма



Слово «алгоритм» появилось в средние века, когда европейцы познакомились со способами выполнения арифметических действий в десятичной системе счисления, описанными узбекским математиком Муххамедом бен Аль-Хорезми («Аль-Хорезми» – человек из города Хорезми). Слово алгоритм – есть результат европейского произношения слов Аль-Хорезми.

# Свойства алгоритмов



# Свойства алгоритмов

- **Дискретность** (разрывность) – это свойство алгоритма, характеризующее его структуру: каждый алгоритм состоит из отдельных законченных действий.
- **Массовость** – применимость алгоритма ко всем задачам рассматриваемого типа, при любых исходных данных.
- **Определенность** – свойство алгоритма, указывающее на то, что каждый шаг алгоритма должен быть строго определен и не допускать различных толкований; также строго должен быть определен порядок выполнения отдельных шагов.
- **Результативность** - конечность действий алгоритма решения задач, позволяющая получить желаемый результат при допустимых исходных данных за конечное число шагов.

# Способы описания алгоритмов

- ▣ **Словесно-формульное** описание алгоритма, т.е. описание алгоритма с помощью слов и формул.
- ▣ **Графическое** описание алгоритма, т.е. описание алгоритма с помощью схем. Схема представляет собой систему связанных геометрических фигур. Каждая фигура обозначает этап процесса решения задачи и называется блоком.
- ▣ **Описание алгоритма на алгоритмическом языке.** Алгоритмический язык – это средство для записи алгоритмов в аналитическом виде, промежуточном между записью алгоритма на естественном языке и записью на языке ЭВМ ( языке программирования).

# Способы описания алгоритмов

**Словесное описание** представляет структуру алгоритма на естественном языке.

**Пример:** инструкция по эксплуатации любого прибора бытовой техники (утюг, телевизор, электрочайник), рецепт блюда, правила дорожного движения.

**Словесная форма имеет ряд недостатков:**

- строго не формализуема;
- страдает многословностью записей;
- допускает неоднозначность толкования отдельных предписаний.

Обычно используется на начальных стадиях разработки алгоритма.



# Способы описания алгоритмов

**Блок-схема** – это наглядное графическое представление алгоритма с помощью геометрических фигур, соединенных линиями связями, показывающими порядок выполнения инструкций.

# Графические объекты блок-схем



Начало

- Начало алгоритма



Конец

- Конец алгоритма



<Действие>

- Процесс



Ввод/Вывод

- Ввод/вывод данных с  
неопределенного носителя

# Графические объекты блок-схем



- Ввод с клавиатуры



- Вывод на монитор

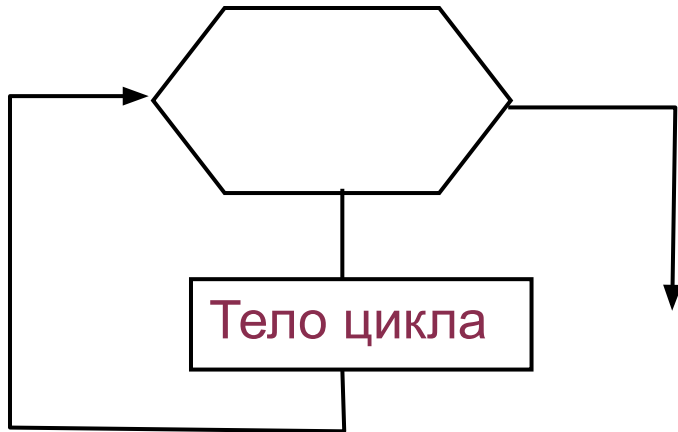


- Вывод на печатающее устройство



- Условный блок

## Графические объекты блок-схем



- Цикл с параметром



Тело цикла

- Границы цикла



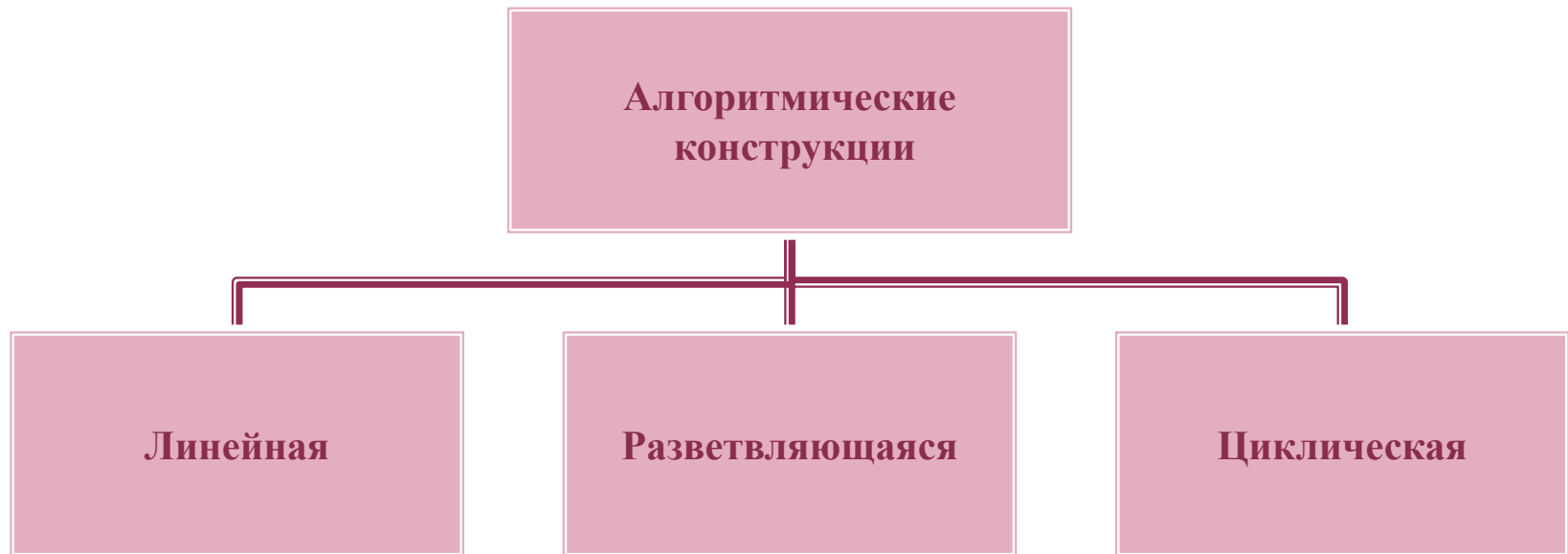
# Способы описания алгоритмов

Программа – описание структуры алгоритма на языке программирования.

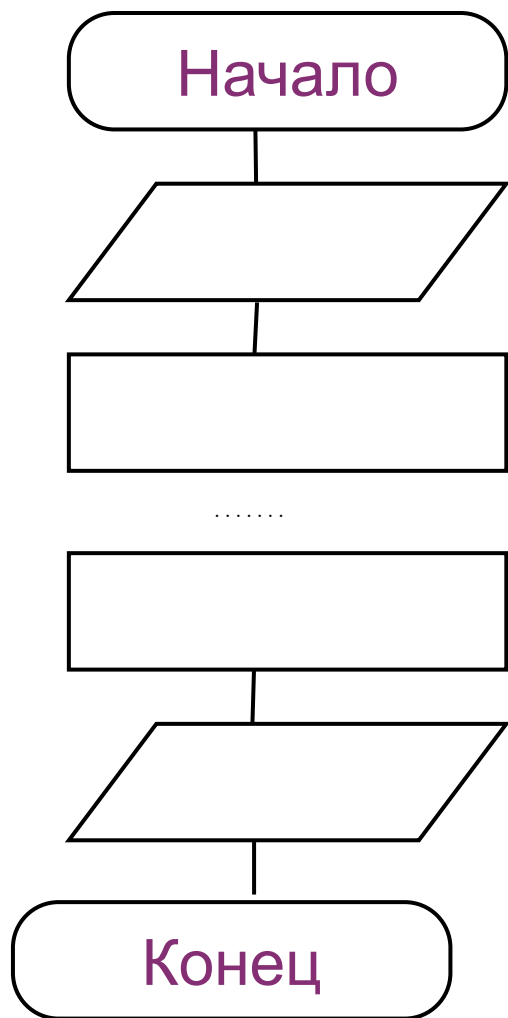
Пример:

```
program sistema ;  
var  
x, xn, xk, dx, y: real ;  
uses crt ;  
begin  
clrscr;  
writeln ('Введите начальное, конечное значение x и  
шаг') ;  
readln (xn, xk,dx) ;  
x:=xn;  
repeat  
if x > 0 then y:=ln(x)  
else y:=sqr(x)+5*x;  
writeln ('x=', x:6:2, 'y=',y) ;  
x:=x+dx;  
until x > xk;  
readln;  
end.
```

# Основные алгоритмические конструкции



# Линейная алгоритмическая конструкция



**Линейный алгоритм** – это описание последовательности действий, которые выполняются однократно в заданном порядке

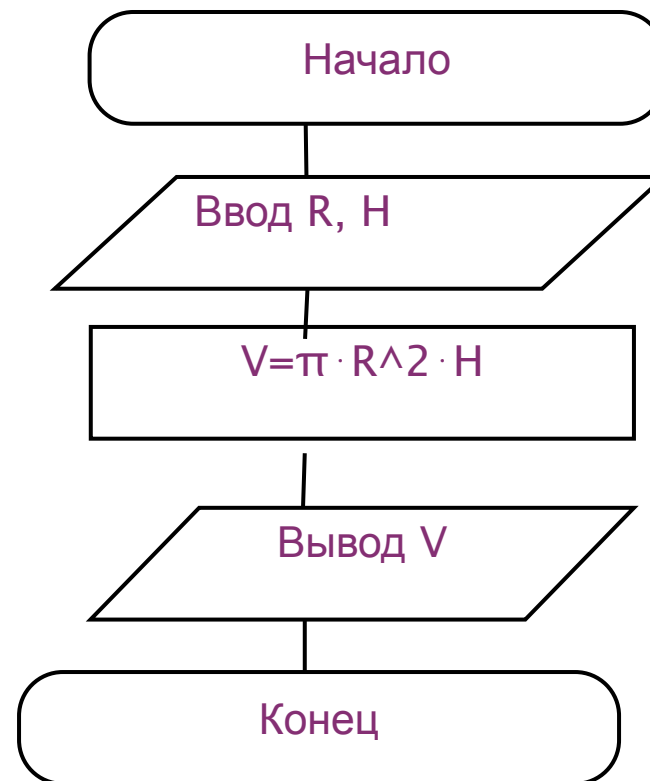
# Примеры линейных алгоритмов



## ПРИМЕР 1

Задача. Зная радиус основания и высоту цилиндра, вычислить его объем.

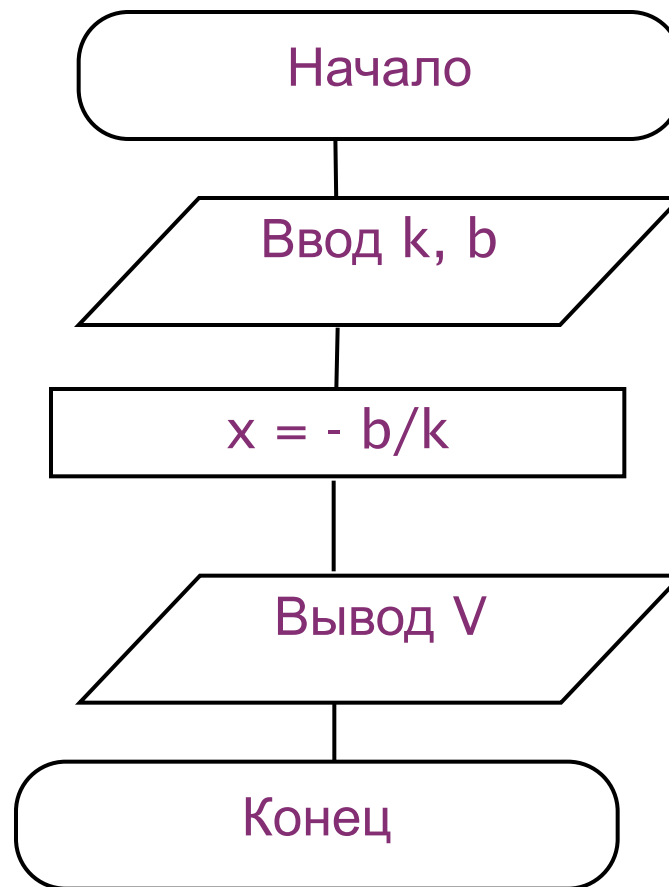
Блок-схема:



## ПРИМЕР 2

Задача. Составить алгоритм для решения линейного уравнения  $k \cdot x + b = 0$ .

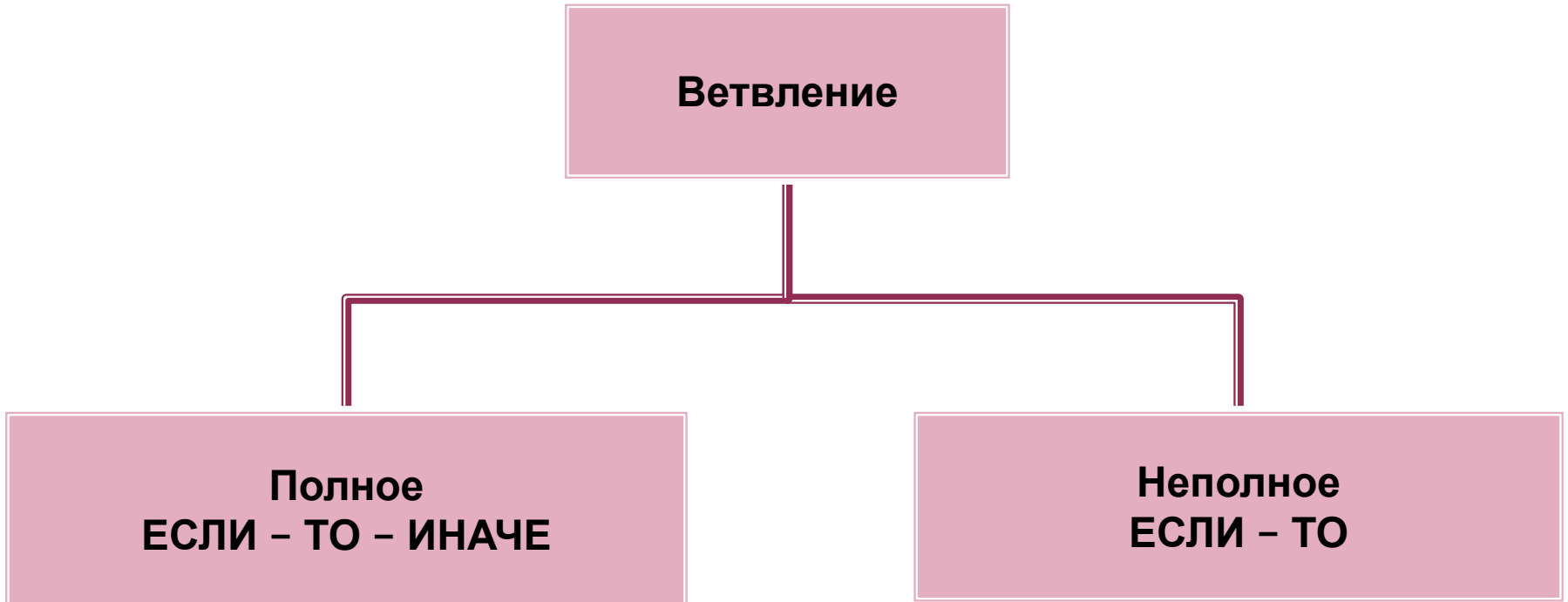
Блок-схема:



# Разветвляющаяся алгоритмическая конструкция

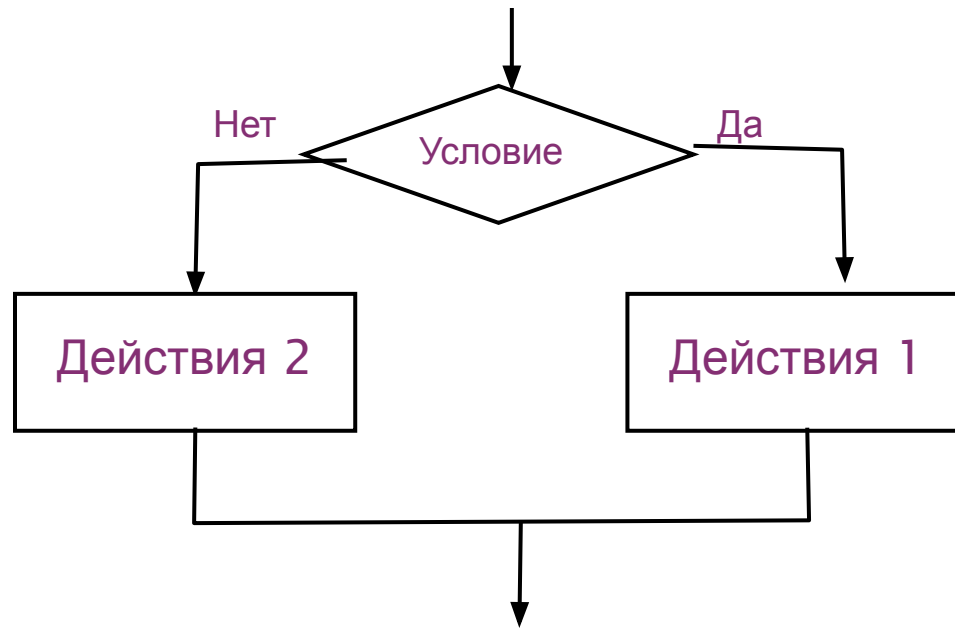
**Разветвляющийся алгоритм** – такой алгоритм, в котором выполняется либо одна, либо другая последовательность действий, в зависимости от условия.

# Разветвляющаяся алгоритмическая конструкция



# Разветвляющаяся алгоритмическая конструкция

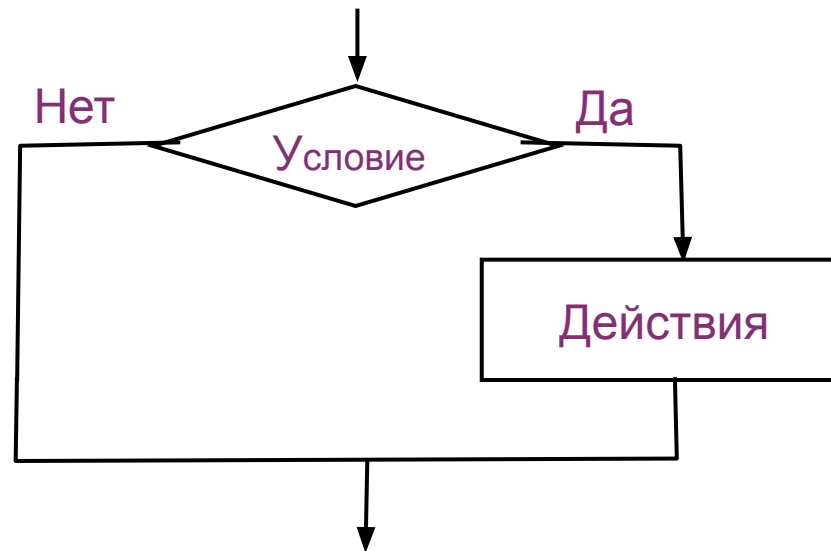
## Полное ветвление



Полное ветвление позволяет организовать в алгоритме две ветви (ТО или ИНАЧЕ).

# Разветвляющаяся алгоритмическая конструкция

## Неполное ветвление

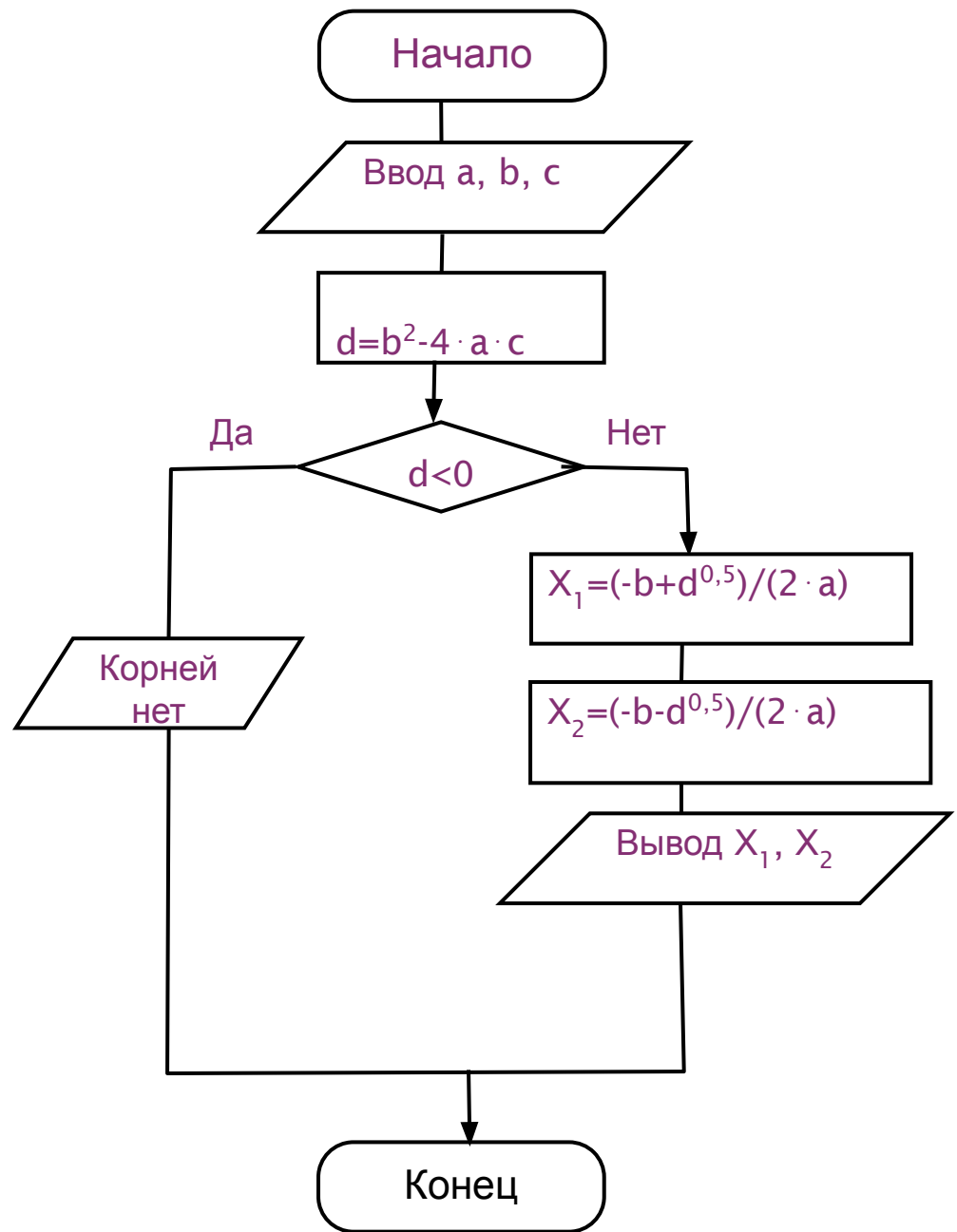


Неполное ветвление предполагает наличие действий только на одной ветви (ТО), вторая ветвь отсутствует.

# ПРИМЕР 1

**ЗАДАЧА.** Известны коэффициенты  $a$ ,  $b$ ,  $c$  квадратного уравнения. Вычислить корни квадратного уравнения.

## Блок-схема:



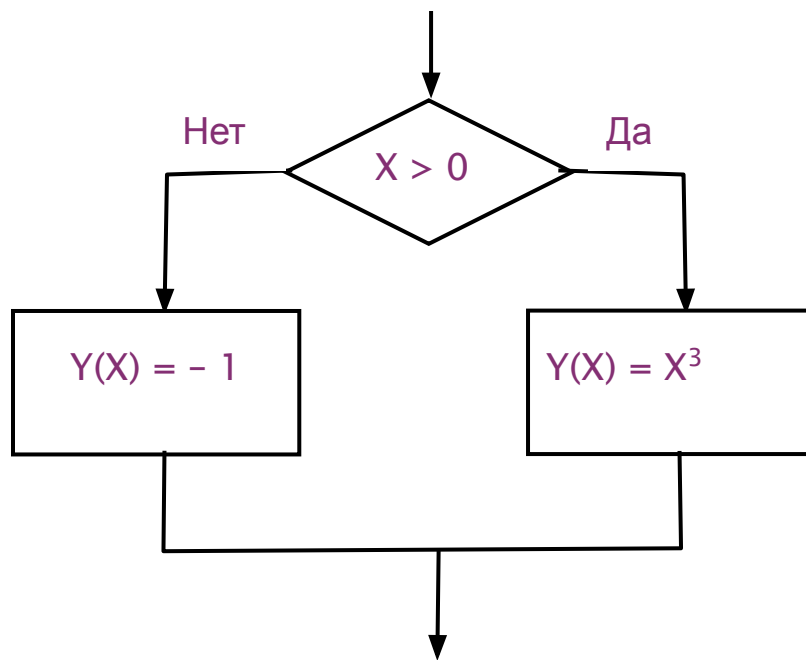


## ПРИМЕР 2

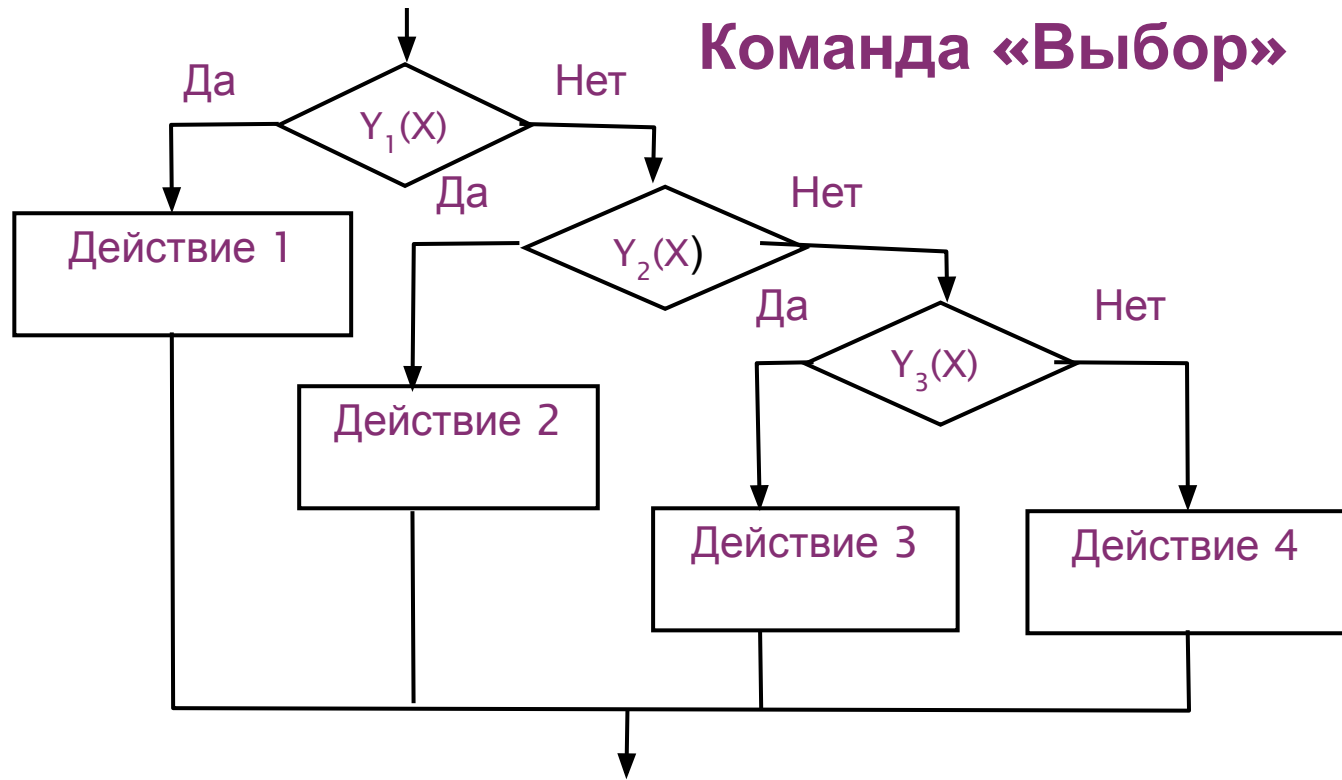
Задача. Составить блок-схему к алгоритму вычисления значения функции

$$Y(X) = \begin{cases} -1, & x \leq 0 \\ x^3, & x > 0 \end{cases}$$

## ПРИМЕР 2

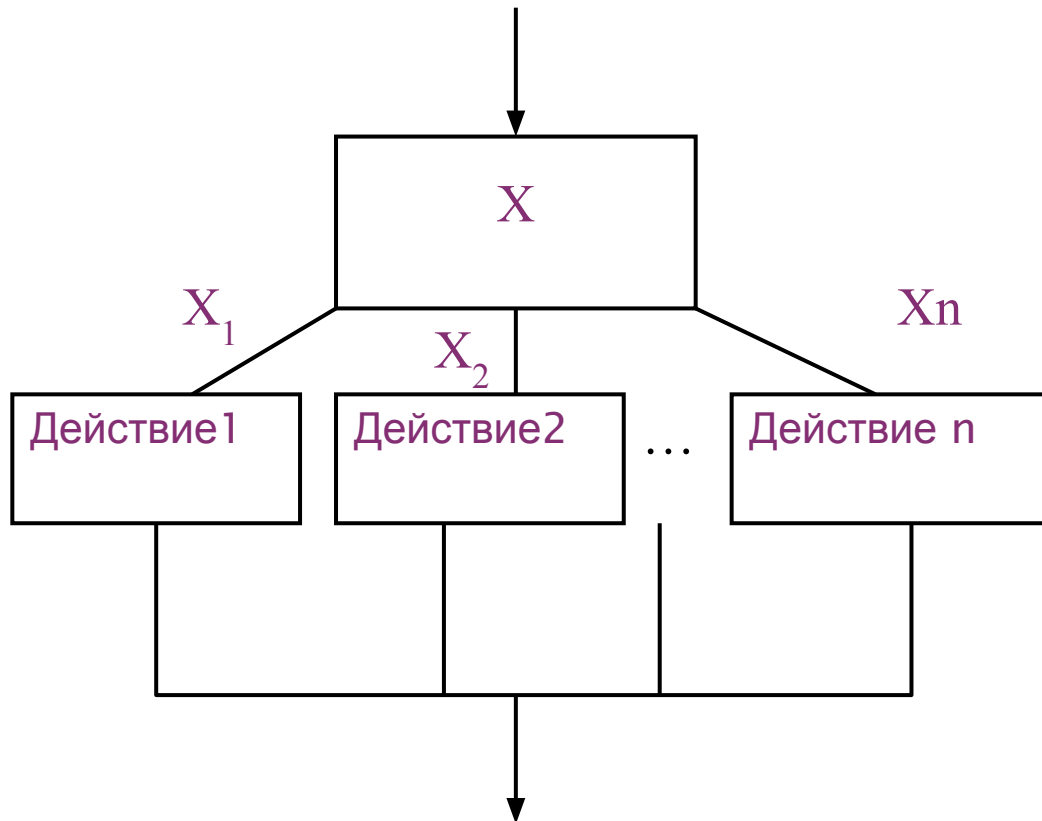


## Команда «Выбор»



Перед выполнением команды «выбор» вычисляется значение некоторого выражения  $X$ , а затем начинается проверка условий  $Y_1(X)$ ,  $Y_2(X)$ ... $Y_n(X)$ . Проверка продолжается до тех пор, пока не встретится условие, принимающее значение ИСТИНА при данном  $X$ .

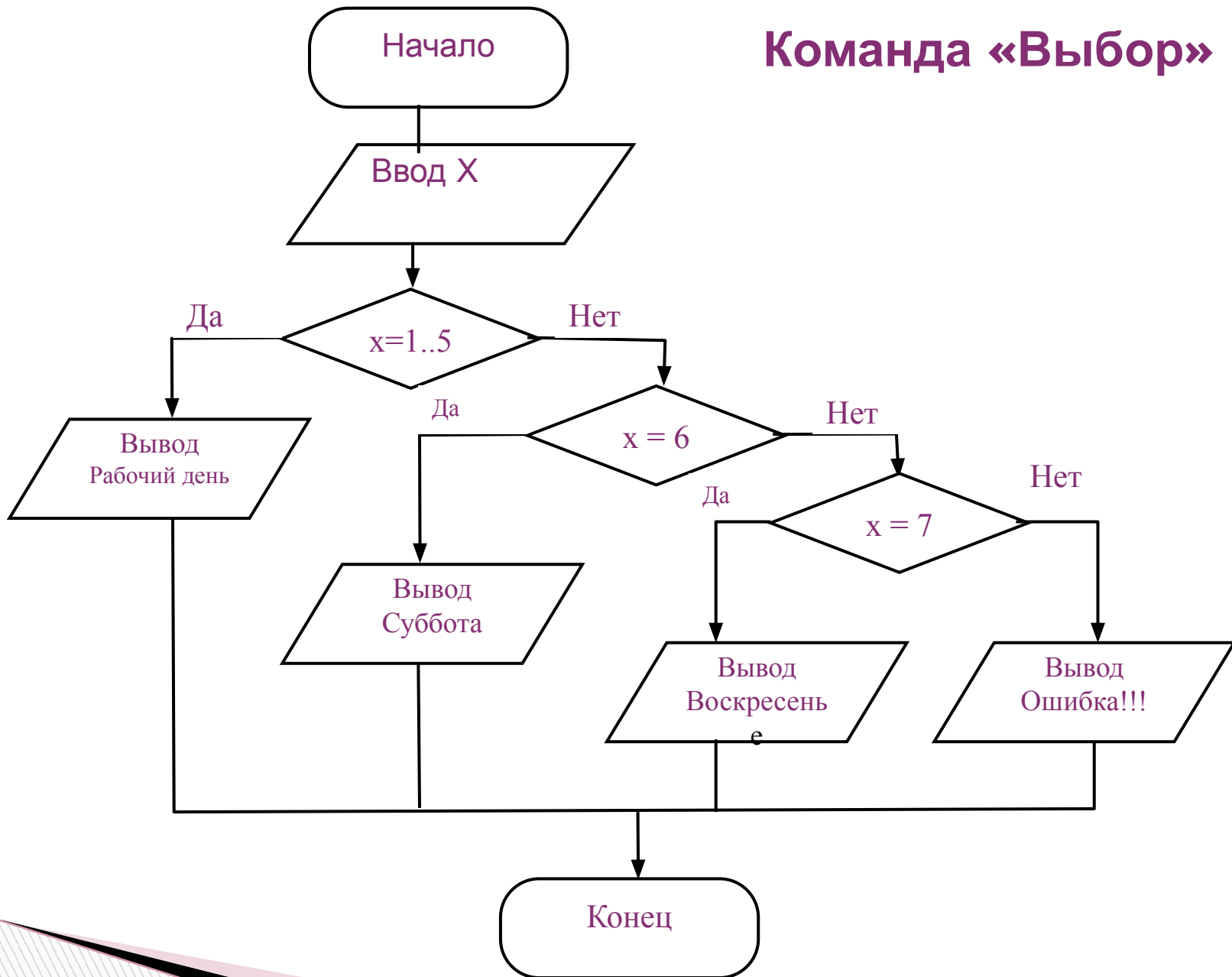
# Команда «Выбор»



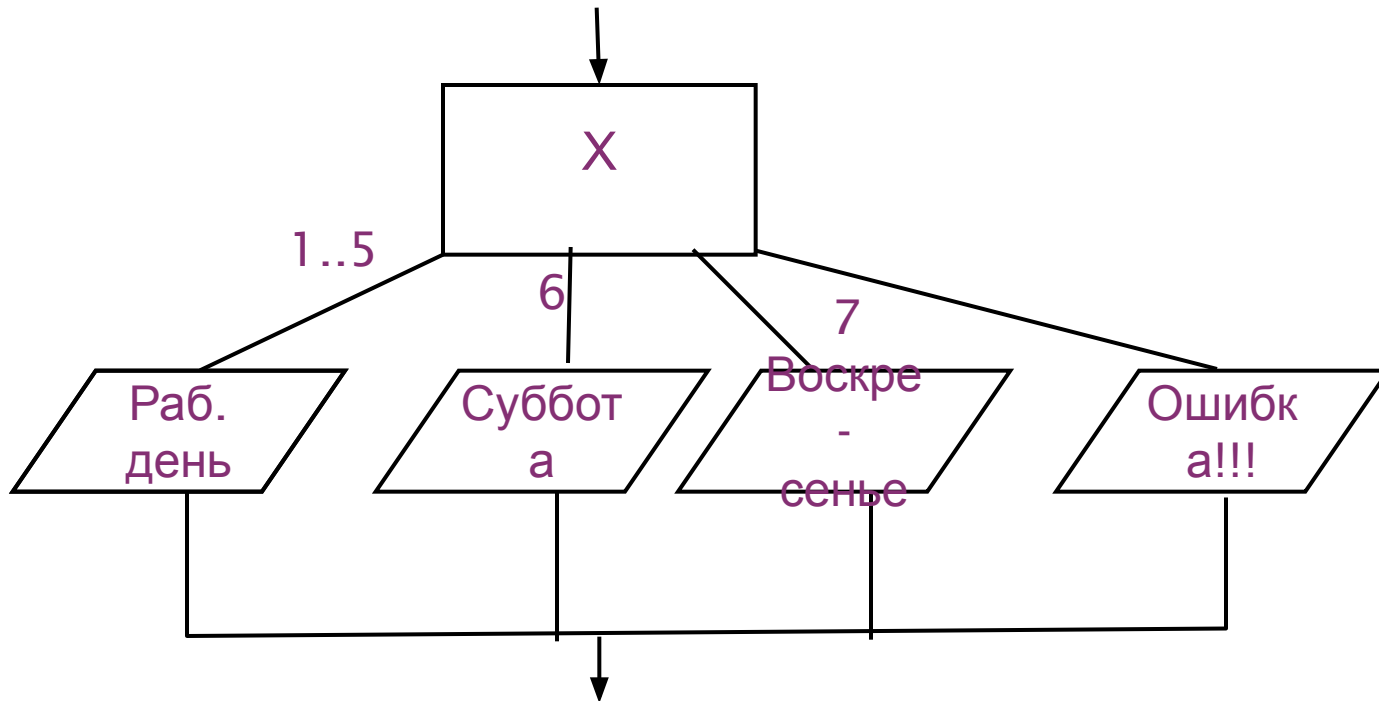
# ПРИМЕР

Задача. Составить блок-схему к программе, которая запрашивает у пользователя номер дня недели и выводит одно из сообщений «Рабочий день», «Суббота» или «Воскресенье».

# Команда «Выбор»



## 2 Вариант



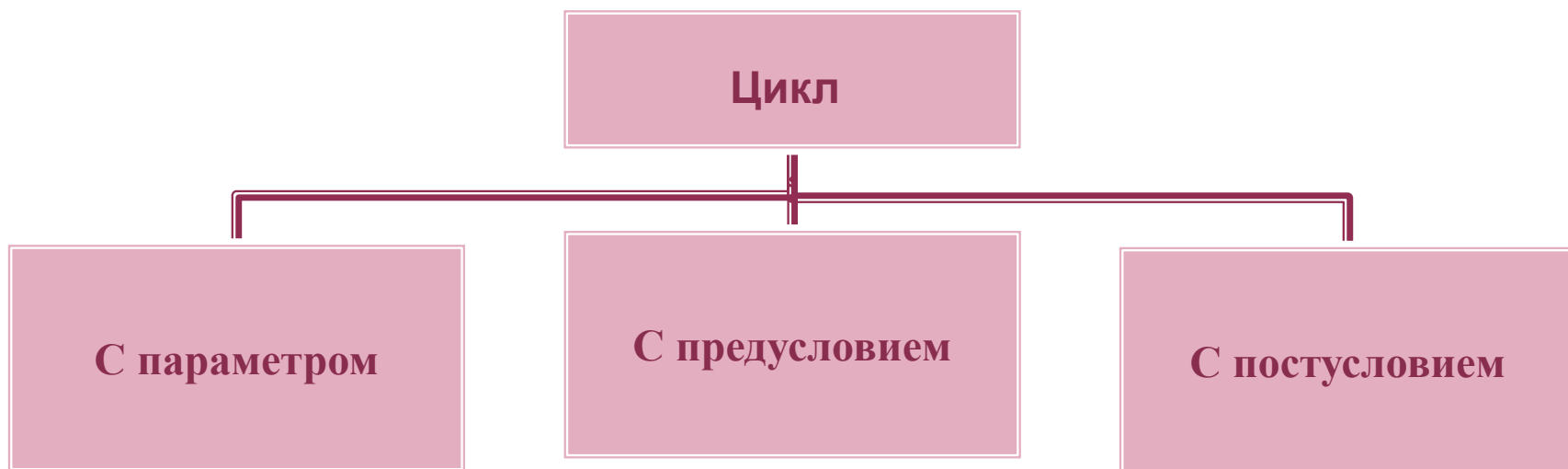
# Циклическая алгоритмическая конструкция

**Циклической** называют алгоритмическую конструкцию, в которой действие выполняется указанное число раз, или, пока не выполнится условие.

Группа повторяющихся действий цикла называется **телом цикла**.

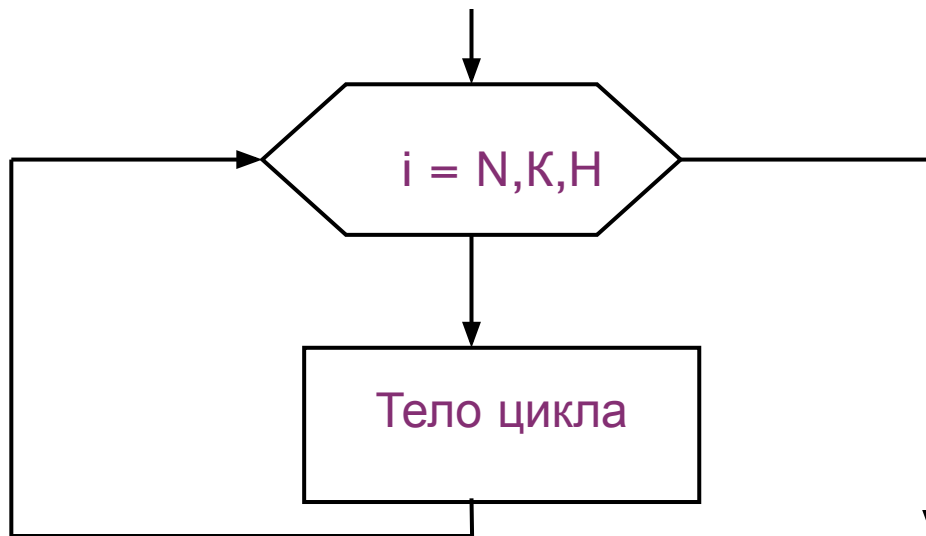


# Циклическая алгоритмическая конструкция



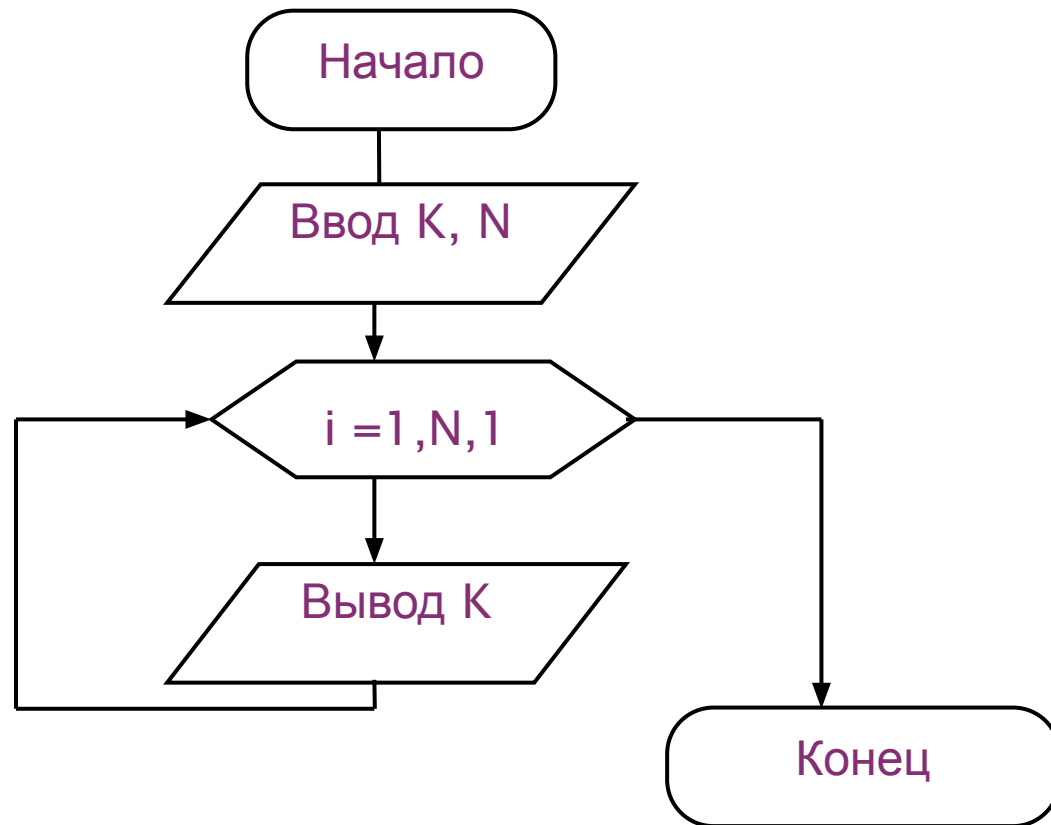
## Цикл с параметром

В цикле с параметром число повторений цикла однозначно определено и задается с помощью начального, конечного значений параметра и шагом его изменения.



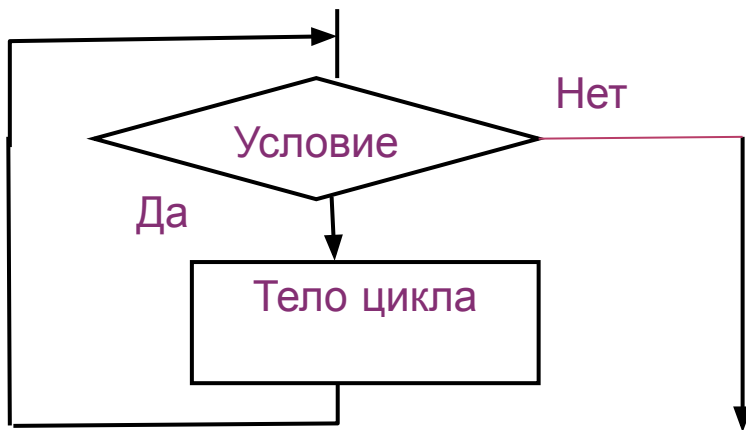
## Цикл с параметром

**ПРИМЕР.** Даны целые числа  $K$  и  $N$  ( $N > 0$ ). Вывести  $N$  раз число  $K$ .



## Цикл с предусловием

Действия внутри этого цикла повторяются, пока выполняется условие в блоке ветвления, причем сначала проверяется условие, а затем выполняется действие.



## Цикл с предусловием

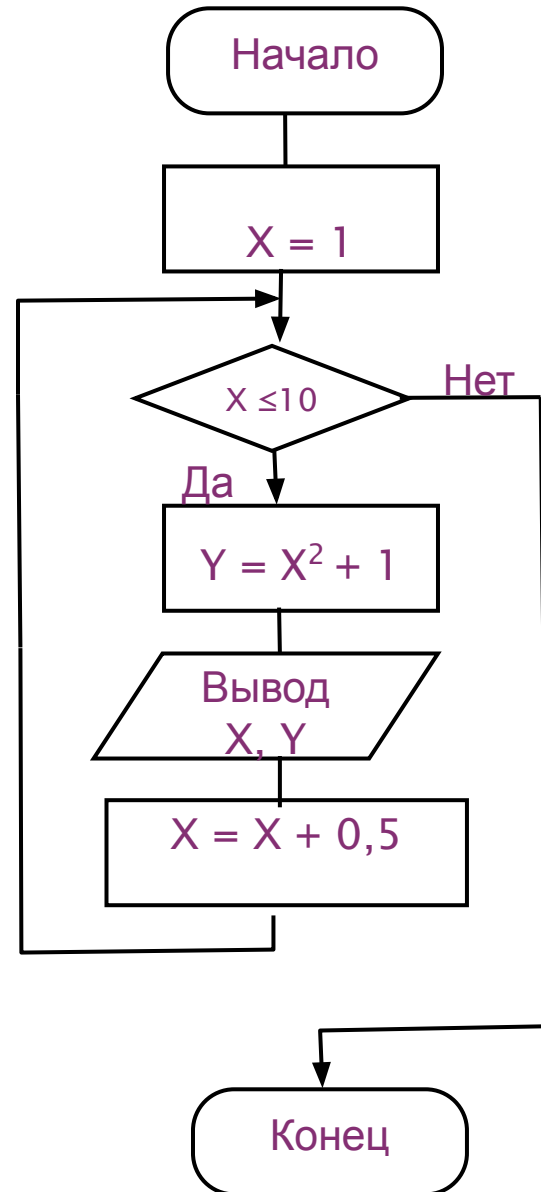
### ПРИМЕР.

Найти значение всех  $Y = X^2 + 1$  при  $X$ , изменяющемся от 1 до 10 с шагом 0,5.

## Цикл с предусловием

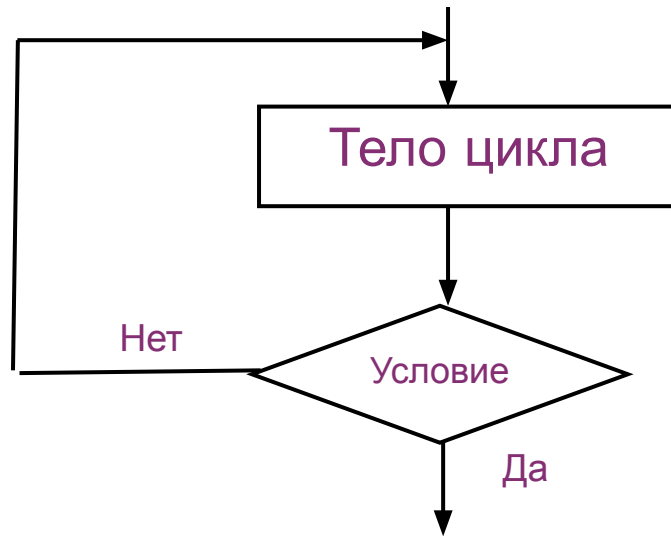
### ПРИМЕР.

Найти значение всех  $Y = X^2 + 1$  при  $X$ , изменяющемся от 1 до 10 с шагом 0,5.



## Цикл с постусловием

Тело цикла с постусловием всегда будет выполнено хотя бы один раз. Оно будет выполняться до тех пор, пока значение условного выражения ЛОЖНО. Как только условное выражение принимает значение ИСТИНА, цикл завершается.



## Цикл с постусловием

**ПРИМЕР.** Составить блок-схему к программе, которая запрашивает у пользователя положительные числа, считает их сумму и количество. Как только введено отрицательное число или ноль, программа завершается.



## Цикл с постусловием

**ПРИМЕР.** Составить блок-схему к программе, которая запрашивает у пользователя положительные числа, считает их сумму и количество. Как только введено отрицательное число или ноль, программа завершается.

Правило суммирования:

1. Необходимо задать начальное значение суммы  $S = 0$ .
2. В теле циклической конструкции выполнить команду:  $S = S + \langle \text{слагаемое} \rangle$ .

## Цикл с постусловием

**ПРИМЕР.** Составить блок-схему к программе, которая запрашивает у пользователя положительные числа, считает их сумму и количество. Как только введено отрицательное число или ноль, программа завершается.

Правило суммирования:

1. Необходимо задать начальное значение суммы  $S = 0$ .
2. В теле циклической конструкции выполнить команду:  $S = S + \langle \text{слагаемое} \rangle$ .

Правило счетчика:

1. Начальное значение счетчика  $K = 0$ .
2. В теле цикла выполнить команду  $K = K + 1$ .

# Блок-схема

