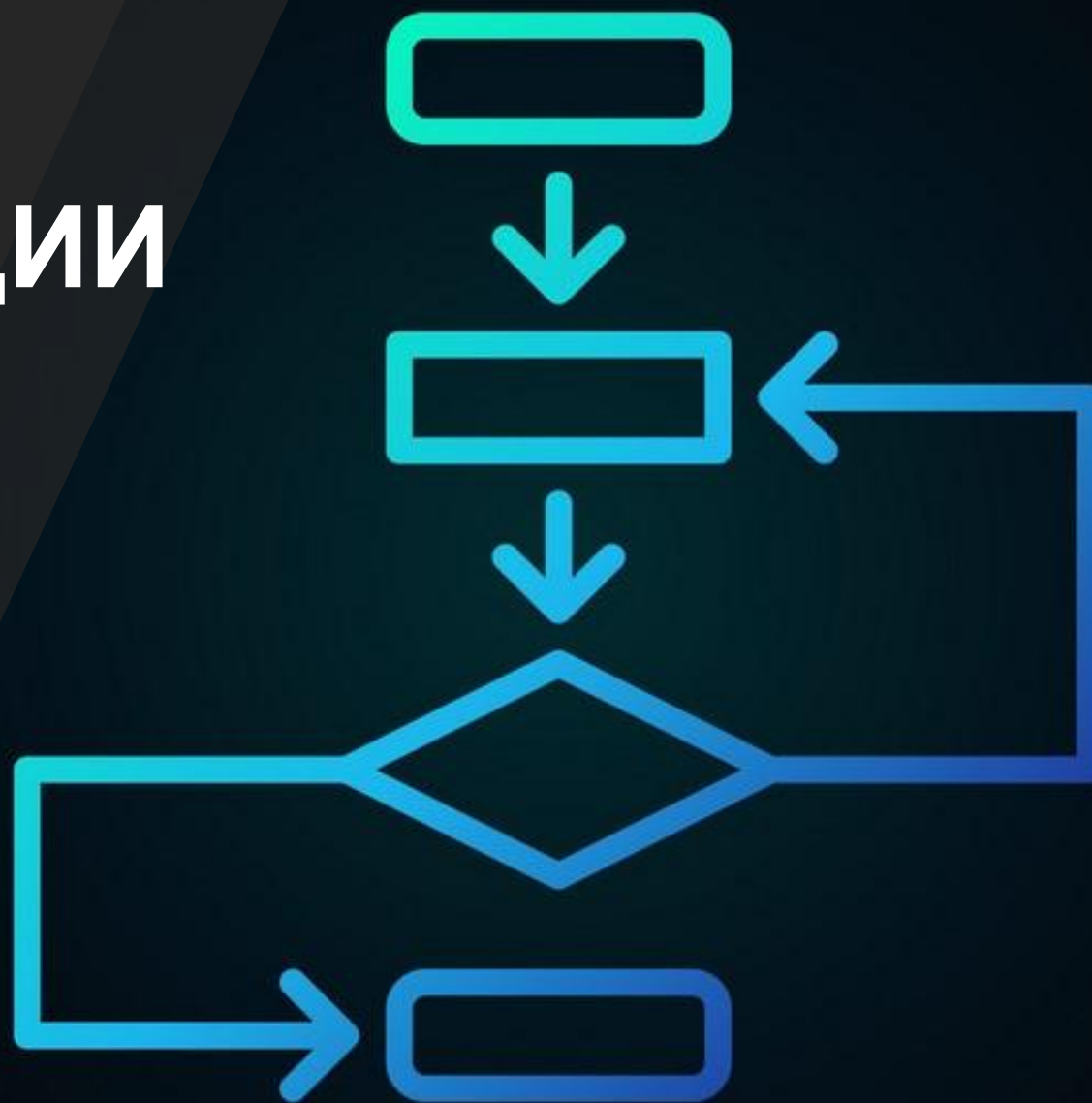


# ОСНОВЫ АЛГОРИТМИЗАЦИИ



Вензовская Наталья  
Юрьевна  
ГБПОУ МО  
«Колледж «Подмосковье»

## 1.1. Понятие алгоритма

Алгоритмом называется строго определенная последовательность действий, определяющих процесс перехода от исходных данных к искомому результату.

## 1.2 Данные и величины.

Совокупность величин, с которыми работает компьютер, принято называть данными. По отношению к программе различают исходные, окончательные (результаты) и промежуточные данные, которые получают в процессе вычислений



Любая величина имеет три основных свойства: имя, значение и тип. На уровне команд процессора величина идентифицируется адресом ячейки памяти, в которой она хранится. В алгоритмах и языках программирования величины подразделяются на константы и переменные.

**Константа** — неизменная величина, и в алгоритме она представляется собственным значением, например: 15, 34.7, k, True и др.

**Переменная величина** — изменяет свое значения в ходе выполнения программы и представляются в алгоритме символическими именами — идентификаторами, например: X, S2, cod15 и др.

Любые константы и переменные занимают ячейку памяти, а значения этих величин определяются двоичным кодом в этой ячейке.

## Основные типы данных

Тип	Значения	Операции	Внутреннее представление
Целые	Целые положительные и отрицательные числа в некотором диапазоне, например: 23, -12, 387	Арифметические операции с целыми числами: сложение, вычитание, умножение, целое деление и деление с остатком. Операции отношений (<, >, = и др.)	Формат с фиксированной точкой
Вещественные	Любые (целые и дробные) числа в некотором диапазоне, например: 2.5, -0.01, 45.0, $3.6 \times 10^9$	Арифметические операции. Операции отношений	Формат с плавающей точкой
Логические	True (истина) False (ложь)	Логические операции: И (and), ИЛИ (or), НЕТ (not). Операции отношений	1 — True; 0 — False
Символьные	Любые символы компьютерного алфавита, например: а, 5, +, \$	Операции отношений	Коды таблицы символьной кодировки, например: ASCII — один символ — 1 байт; Unicode — один символ — 2 байта

## Свойства алгоритма

**Дискретность.** Алгоритм должен представлять процесс решения задачи как последовательность выполнения простых действий (шагов, этапов). При этом для выполнения каждого действия алгоритма требуется время.

**Детерминированность (Однозначность).** Каждое действие (шаг, этап) должно быть четким, однозначным, исключающим произвольное толкование и не оставляющим места для двусмысленности. Выполнение алгоритма носит, по сути, механический характер и не требует никаких дополнительных указаний.

**Массовость.** Алгоритм разрабатывается в общем виде так, чтобы его можно было применять для класса задач, различающихся только исходными данными. При этом исходные данные выбираются из некоторой области, которая называется областью применимости алгоритма.

**Конечность.** Каждое отдельное действие, как и весь алгоритм должны иметь возможность реального исполнения. Поэтому алгоритм имеет предел, т. е. конечен.

**Результативность.** Алгоритм должен приводить к решению задачи или сообщению, что задача решений не имеет за конечное число шагов.

### Способы записи алгоритмов:

- словесно-формульный,
- графический,
- операторный (программа на алгоритмическом языке).

а) *Словесно-формульный способ*. Например, требуется решить квадратное уравнение  $ax^2+bx+c=0$  в области действительных чисел. Математической моделью этой задачи является известная формула корней квадратного уравнения:

$$y_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

На основании этой формулы запишем алгоритм:

1. Задать значения  $a$ ,  $b$ ,  $c$ .
2. Вычислить дискриминант  $d = b^2 - 4ac$ .
3. Сравнить дискриминант с нулем, если он больше нуля, то вычислить корни по формуле

$$y_{1,2} = \frac{-b \pm \sqrt{d}}{2a}$$

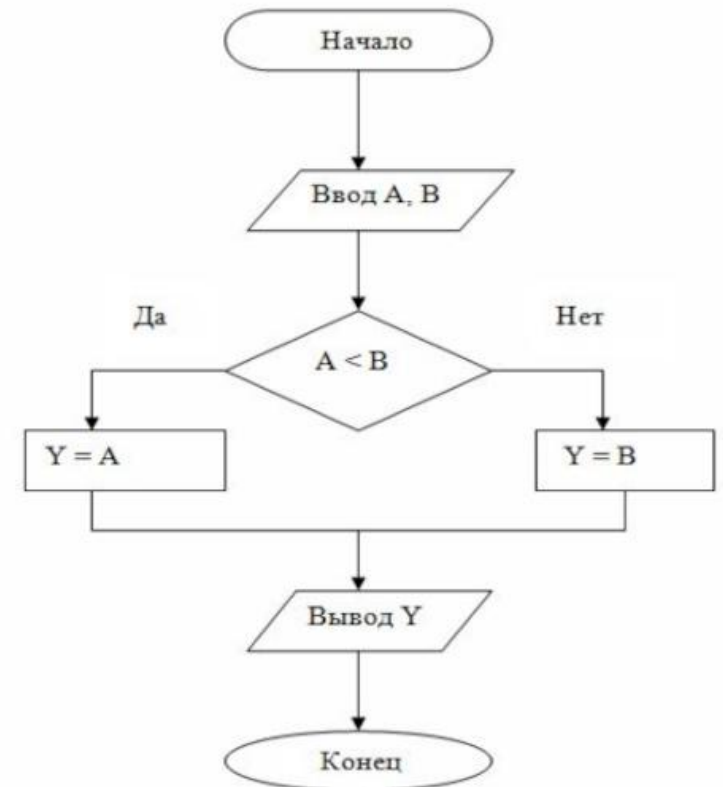
и перейти к п. 4, иначе сообщить: В области действительных чисел уравнение решений не имеет.

4. Записать результат: Корни уравнения  $y_1$  и  $y_2$

б) *Графический способ описания алгоритма* иначе называют блок - схемой. В блок-схемах используются геометрические фигуры, каждая из которых изображает какую-либо операцию или действие, а также этап процесса решения задачи. Каждая фигура называется блоком. Порядок выполнения этапов показывается стрелками, соединяющими блоки. Блоки необходимо размещать сверху вниз или слева направо в порядке их выполнения.

### Правила построения алгоритмов на языке блок-схем

1. Блок-схема строится сверху вниз.
2. В любой блок-схеме имеется один элемент, соответствующий началу, и один элемент, соответствующий концу.
3. Должен быть хотя бы один путь из начала блок-схемы к любому элементу.
4. Должен быть хотя бы один путь от каждого элемента блок-схемы в конец блок-схемы.



# Элементы блок-схемы алгоритма



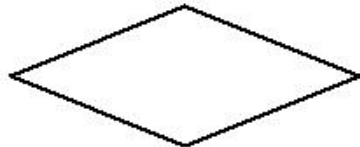
- начало и конец алгоритма



- ВВОД И ВЫВОД ДАННЫХ



- последовательность команд



- условие



- счетчик



- объявление переменных

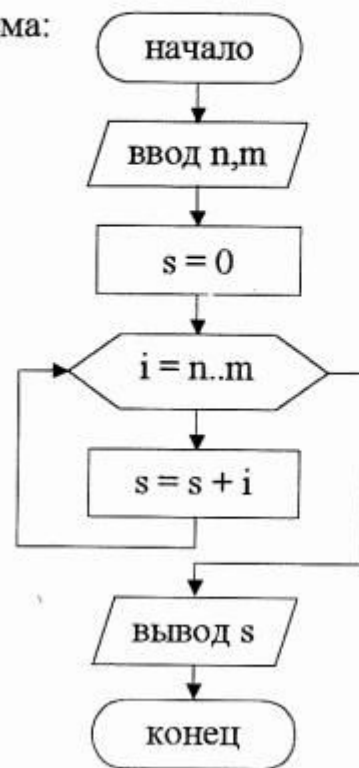
в) *Операторный способ (алгоритмический язык).*

Самые известные алгоритмические языки – это Бейсик (Basic), Паскаль (Pascal), Фортран (Fortran).

Алгоритмический язык отличается от машинного языка тем, что состоит из слов и символов, как естественный язык.

Алгоритмический язык отличается от естественного языка тем, что в нем мало основных слов (обычно 30-40) и очень строгие правила составления предложений. Основные слова алгоритмического языка называют служебными словами. В алгоритмических языках используют слова английского алфавита. Алгоритмический язык легко понимает и человек и компьютер.

Блок-схема:



Программа:

```
Program Summa;  
Var  
  n,m,i,s:integer;  
Begin  
  Write('n=');Readln(n);  
  Write('m=');Readln(m);  
  s:=0;  
  For i:= n to m do s:=s+i;  
  Writeln('s=',s);  
  Readln;  
End.
```