

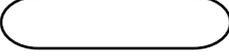
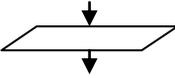
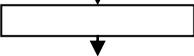
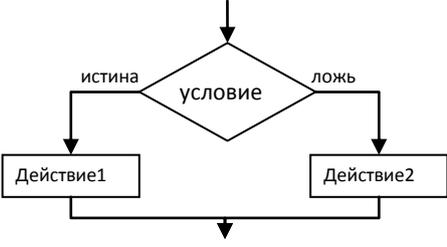
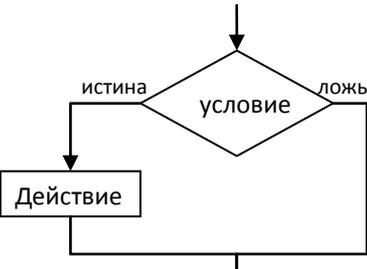
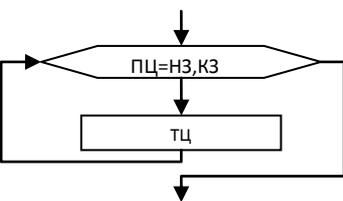
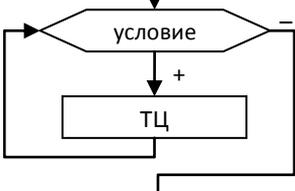
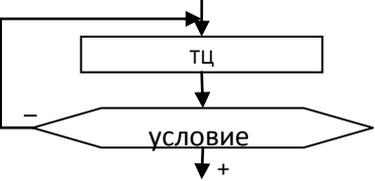
Язык блок-схем, как способ записи алгоритмов. Виды алгоритмов.

При *блок-схемном* описании алгоритм изображается геометрическими фигурами (блоками), связанными по управлению линиями со стрелками. В блоках записывается последовательность действий.

Данный способ по сравнению с другими способами записи алгоритма имеет ряд преимуществ. Он наиболее нагляден: каждая операция вычислительного процесса изображается отдельной геометрической фигурой. Кроме того, графическое изображение алгоритма наглядно показывает разветвления путей решения задачи в зависимости от различных условий, повторение отдельных этапов вычислительного процесса и Другие детали.

В пределах одной схемы рекомендуется изображать блоки одинаковых размеров. Все блоки нумеруются.

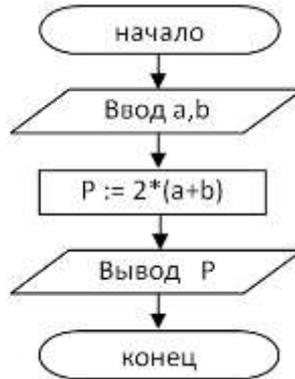
В следующей таблице приведены основные блоки и их обозначения.

	<p><i>Начало/Конец</i> алгоритма</p>
	<p>Блок <i>ввода данных</i> с клавиатуры/<i>вывода данных</i> на экран</p>
	<p>Блок <i>действий</i></p>
	<p><i>Условный оператор</i> (полная конструкция)</p>
	<p><i>Условный оператор</i> (неполная конструкция)</p>
	<p>Оператор цикла (<i>цикл с параметром</i>) ПЦ – параметр цикла НЗ – начальное значение КЗ – конечное значение ТЦ – тело цикла</p>
	<p>Оператор цикла (<i>цикл с предусловием</i>) ТЦ – тело цикла</p>
	<p>Оператор цикла (<i>цикл с постусловием</i>) ТЦ – тело цикла</p>

Виды алгоритмов

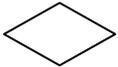
1. **Линейные алгоритмы** – алгоритмы, в которых все действия (блоки) выполняются последовательно, друг за другом. При выполнении алгоритма имеется только один способ его прохождения.

Пример: Вычислить периметр прямоугольника по двум сторонам.



2. **Алгоритмы с ветвлением** - алгоритмы, в которых в зависимости от условия выполняется либо одна, либо другая последовательность действий.

Алгоритмы с ветвлением отличаются от линейных алгоритмов наличием блока «ромб», обозначающего условный оператор.



Условные операторы бывают двух видов:

- в полной форме

Например: если число делится на 2, то оно четное, иначе число – нечетное.

Обозначение этого оператора в блок-схеме смотри в таблице.

- в неполной форме

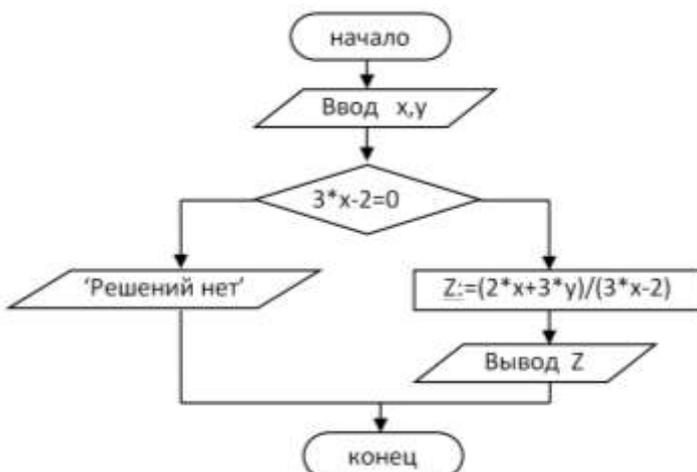
Например: если на улице дождь, то необходимо взять зонт.

Обозначение этого оператора в блок-схеме смотри в таблице.

Пример: Вычислить значение выражения:

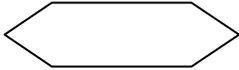
$$z = \frac{2x + 3y}{3x - 2}$$

Вычислить значение Z можно, когда знаменатель дроби не равен нулю. Поэтому алгоритм имеет два варианта решения: при $3x-2=0$ - решений нет, а в противном случае – необходимо вычислить Z и вывести результат.



3. **Циклические алгоритмы** - описание действий, которые должны повторяться указанное число раз или по заданному условию. Перечень повторяющихся действий называется **телом цикла**.

Циклические алгоритмы отличаются от линейных алгоритмов и алгоритмов с ветвлением наличием блока «шестиугольник», обозначающего условный оператор.



Условные операторы бывают двух видов:

- Цикл с параметром (со счетчиком) или арифметический цикл

Например: Необходимо измерить рост всех учащихся в классе. Если известно, что в классе 20 человек, то берем первого ученика — меряем его рост и записываем, берем следующего ученика и т. д., до последнего 20 учащегося.

Обозначение этого оператора в блок-схеме смотри в таблице.

- Цикл по условию

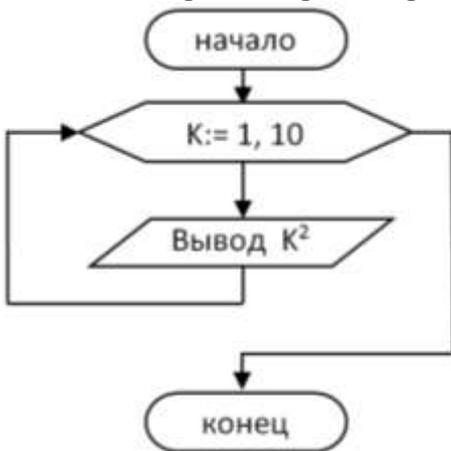
Например: колка дров. Если количество поленьев не известно, берем полено — колем топором, берем второе полено и т. д., пока поленья не закончатся.

Циклы по условию в свою очередь делятся на циклы с предусловием и циклы с пост условием.

Обозначения этих операторов в блок-схеме смотри в таблице.

Пример №1:

Вывести на экран квадраты первых десяти натуральных чисел.



Пример №2:

Вычислить остаток от деления числа X на 3 не используя операцию деления.

