

# Алгоритм





# Оглавление

1. Понятие алгоритма
2. Формальные признаки алгоритмов
3. Формальные свойства алгоритмов
4. Виды алгоритмов
5. Изображение алгоритма в виде блок-схемы

# Понятие алгоритма

**Алгоритм** - четкое описание последовательности действий, которые необходимо выполнить при решении задачи. Можно сказать, что алгоритм описывает процесс преобразования исходных данных в результаты, т.к. для решения любой задачи необходимо:

- Ввести исходные данные.
- Преобразовать исходные данные в результаты (выходные данные).
- Вывести результаты.



# Понятие алгоритма

«Алгоритм - это всякая система вычислений, выполняемых по строго определённым правилам, которая после какого - либо числа шагов заведомо приводит к решению поставленной задачи.»

(А. Колмогоров)



# Понятие алгоритма

«Алгоритм - однозначно, доступно и кратко (условные понятия - названия этапа) описанная последовательность процедур для воспроизводства процесса с обусловленным задачей алгоритма результатом при заданных начальных условиях.

Универсальность (или специализация) алгоритма определяется применимостью и надёжностью данного алгоритма для решения нестандартных задач.»



# Понятие алгоритма

«Алгоритм — это система операторов, взятых из множества операторов некоторого исполнителя, которая полностью определяет некоторый класс алгоритмических процессов, то есть процессов, которые:

- дискретны;
- детерминированы;
- потенциально конечны;
- преобразовывают некоторые конструктивные объекты.

Между операторами алгоритма и операциями (элементарными действиями) алгоритмического процесса существует гомоморфное соответствие. Поэтому алгоритм следует также считать моделью алгоритмического процесса».

(А. Копяев)



# Понятие алгоритма

«Алгоритм - это последовательность действий, направленных на получение определённого результата.»



# Формальные признаки алгоритмов

Различные определения алгоритма в явной или неявной форме содержат следующий ряд общих требований:

- Детерминированность;
- Понятность;
- Завершаемость (конечность);
- Массовость.

Алгоритмы обычно накладывают ограничения на тип исходных данных.



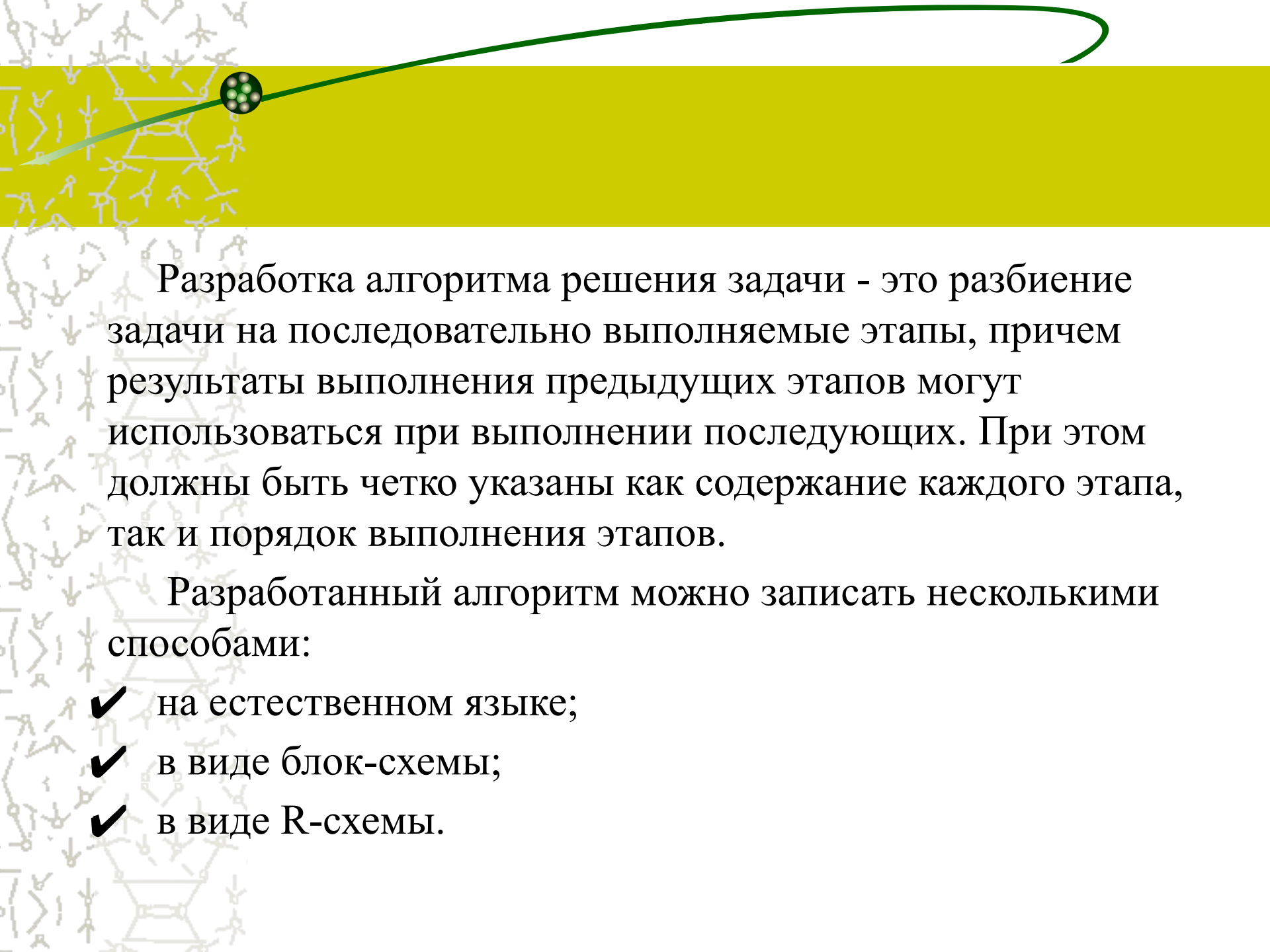
# Формальные свойства алгоритмов

Различные определения алгоритма в явной или неявной форме содержат следующий ряд общих требований:

- ❖ Дискретность;
- ❖ Детерминированность (определённость);
- ❖ Понятность;
- ❖ Завершаемость (конечность);
- ❖ Массовость (универсальность);
- ❖ Результативность;
- ❖ Алгоритм содержит ошибки, если приводит к получению неправильных результатов либо не даёт результатов вовсе.
- ❖ Алгоритм не содержит ошибок, если он даёт правильные результаты для любых допустимых исходных данных.

# Виды алгоритмов

- ✓ Механические алгоритмы, или иначе детерминированные, жесткие.
- ✓ Гибкие алгоритмы.
- ✓ Вероятностный (стохастический) алгоритм.
- ✓ Эвристический алгоритм (от греческого слова «эврика»).
- ✓ Линейный алгоритм.
- ✓ Разветвляющийся алгоритм.
- ✓ Циклический алгоритм.
- ✓ Вспомогательный (*подчиненный*) алгоритм.
- ✓ Структурная блок-схема, граф-схема алгоритма.



Разработка алгоритма решения задачи - это разбиение задачи на последовательно выполняемые этапы, причем результаты выполнения предыдущих этапов могут использоваться при выполнении последующих. При этом должны быть четко указаны как содержание каждого этапа, так и порядок выполнения этапов.

Разработанный алгоритм можно записать несколькими способами:

- ✓ на естественном языке;
- ✓ в виде блок-схемы;
- ✓ в виде R-схемы.

## Пример алгоритма на естественном языке

1. Ввести в компьютер числовые значения переменных **a**, **b** и **c**.
2. Вычислить **d** по формуле  $d = b^2 - 4ac$ .
3. Если  $d < 0$ , то напечатать сообщение "Корней нет" и перейти к п.4. Иначе вычислить и напечатать значения  $x_1$  и  $x_2$ .
4. Прекратить вычисления.

# Изображение алгоритма в виде блок-схемы

**Блок-схемой** называется наглядное графическое изображение алгоритма, когда отдельные его этапы изображаются при помощи различных геометрических фигур - блоков, а связи между этапами (последовательность выполнения этапов) указываются при помощи стрелок, соединяющих эти фигуры.

Блоки сопровождаются надписями.



## Блок начало-конец (пуск-остановка)



Элемент отображает выход во внешнюю среду и вход из внешней среды (наиболее частое применение – начало и конец программы). Внутри фигуры записывается соответствующее действие.

# Блок действия

Выполнение одной или нескольких операций, обработка данных любого вида (изменение значения данных, формы представления, расположения).  
Внутри фигуры записывают непосредственно сами операции, например, операцию присваивания:  $a = 10 * b + c$ .



# Логический блок (блок условия)

Отображает решение или функцию переключательного типа с одним входом и двумя или более альтернативными выходами, из которых только один может быть выбран после вычисления условий, определенных внутри этого элемента. Вход в элемент обозначается линией, входящей обычно в верхнюю вершину элемента. Если выходов два или три, то обычно каждый выход обозначается линией, выходящей из оставшихся вершин (боковых и нижней). Если выходов больше трех, то их следует показывать одной линией, выходящей из вершины (чаще нижней) элемента, которая затем разветвляется. Соответствующие результаты вычислений могут записываться рядом с линиями, отображающими эти пути. Примеры решения: в общем случае – сравнение (три выхода:  $>$ ,  $<$ ,  $=$ ); в программировании – условные операторы `if` (два выхода: `true`, `false`) и `case` (множество выходов).





# Предопределённый процесс

Символ отображает выполнение процесса, состоящего из одной или нескольких операций, который определен в другом месте программы (в подпрограмме, модуле). Внутри символа записывается название процесса и передаваемые в него данные. Например, в программировании – вызов процедуры или функции.



# Данные (ввод-вывод)

Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод). Данный символ не определяет носителя данных (для указания типа носителя данных используются специфические символы).



# Граница цикла

Символ состоит из двух частей – соответственно, начало и конец цикла – операции, выполняемые внутри цикла, размещаются между ними. Условия цикла и приращения записываются внутри символа начала или конца цикла – в зависимости от типа организации цикла. Часто для изображения на блок-схеме цикла вместо данного символа используют символ условия, указывая в нём решение, а одну из линий выхода замыкают выше в блок-схеме (перед операциями цикла).



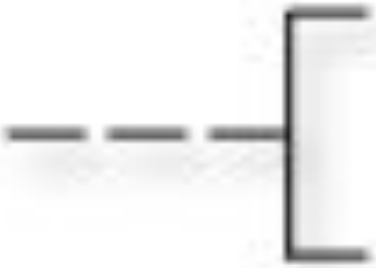
# Соединитель

Символ отображает вход в часть схемы и выход из другой части этой схемы. Используется для обрыва линии и продолжения её в другом месте (для избежания излишних пересечений или слишком длинных линий, а также, если схема состоит из нескольких страниц).

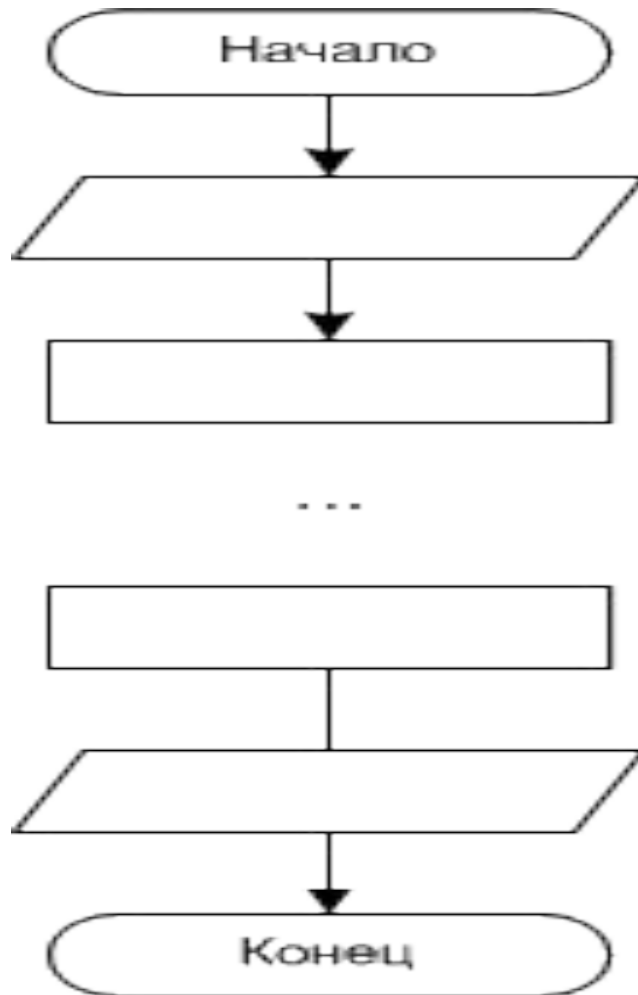
Соответствующие соединительные символы должны иметь одинаковое (при том уникальное) обозначение.

# Комментарий

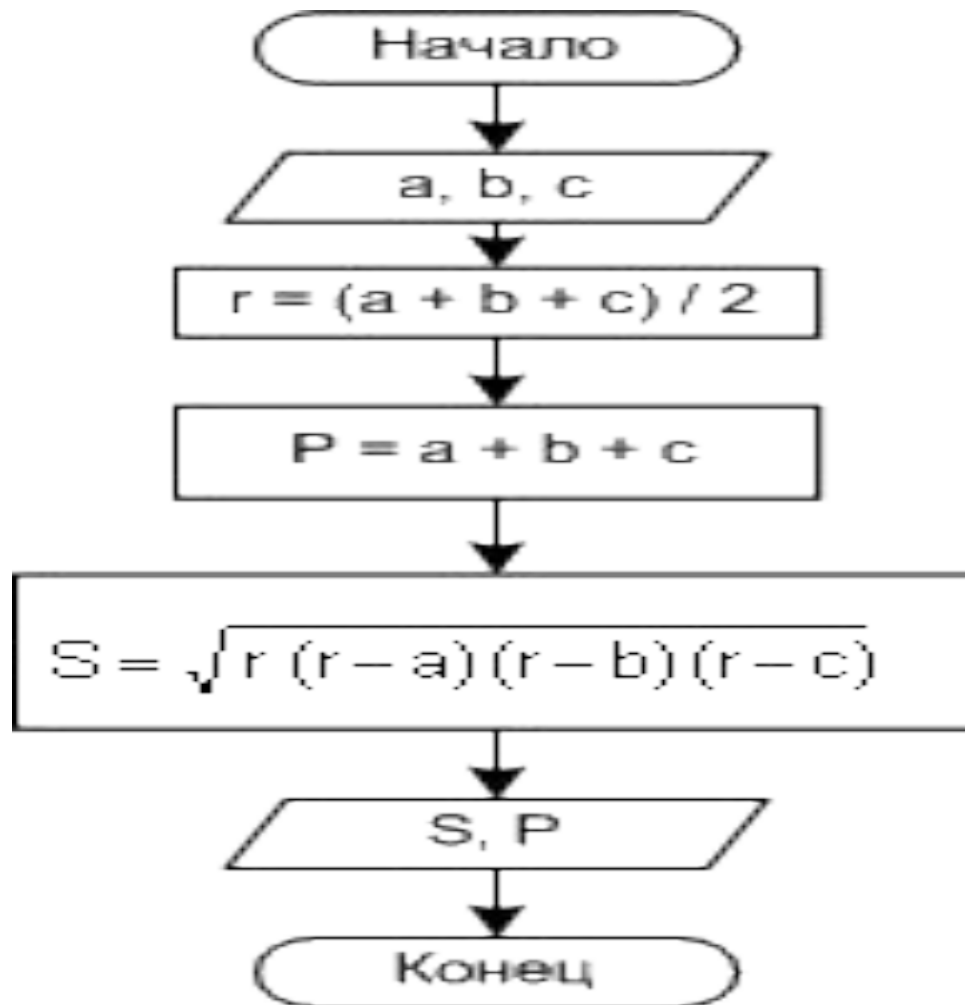
Используется для более подробного описания шага, процесса или группы процессов. Описание помещается со стороны квадратной скобки и охватывается ей по всей высоте. Пунктирная линия идет к описываемому элементу, либо группе элементов (при этом группа выделяется замкнутой пунктирной линией). Также символ комментария следует использовать в тех случаях, когда объём текста, помещаемого внутри некоего символа (например, символ процесса, символ данных и др.), превышает размер самого этого символа.



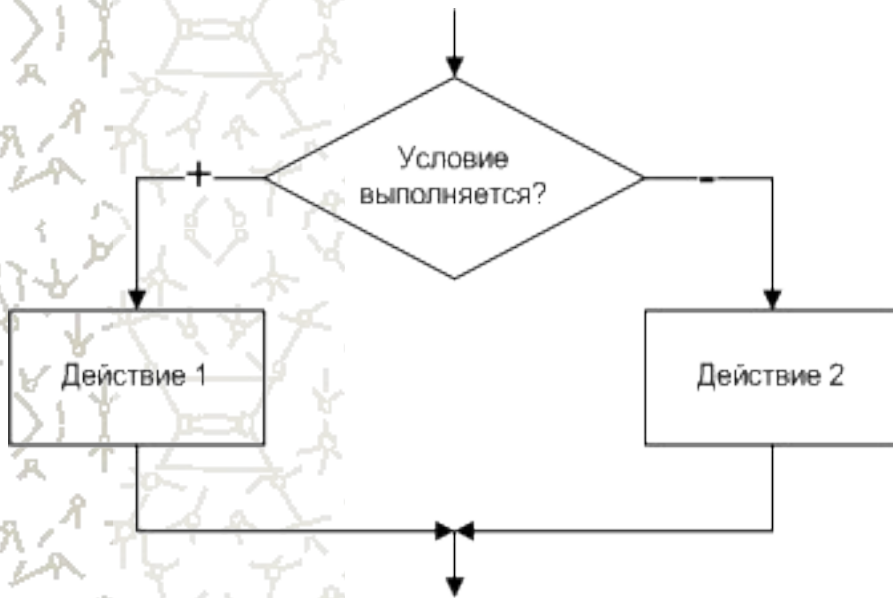
**Линейный алгоритм** - это такой, в котором все операции выполняются последовательно одна за другой.



# БЛОК-СХЕМА АЛГОРИТМА

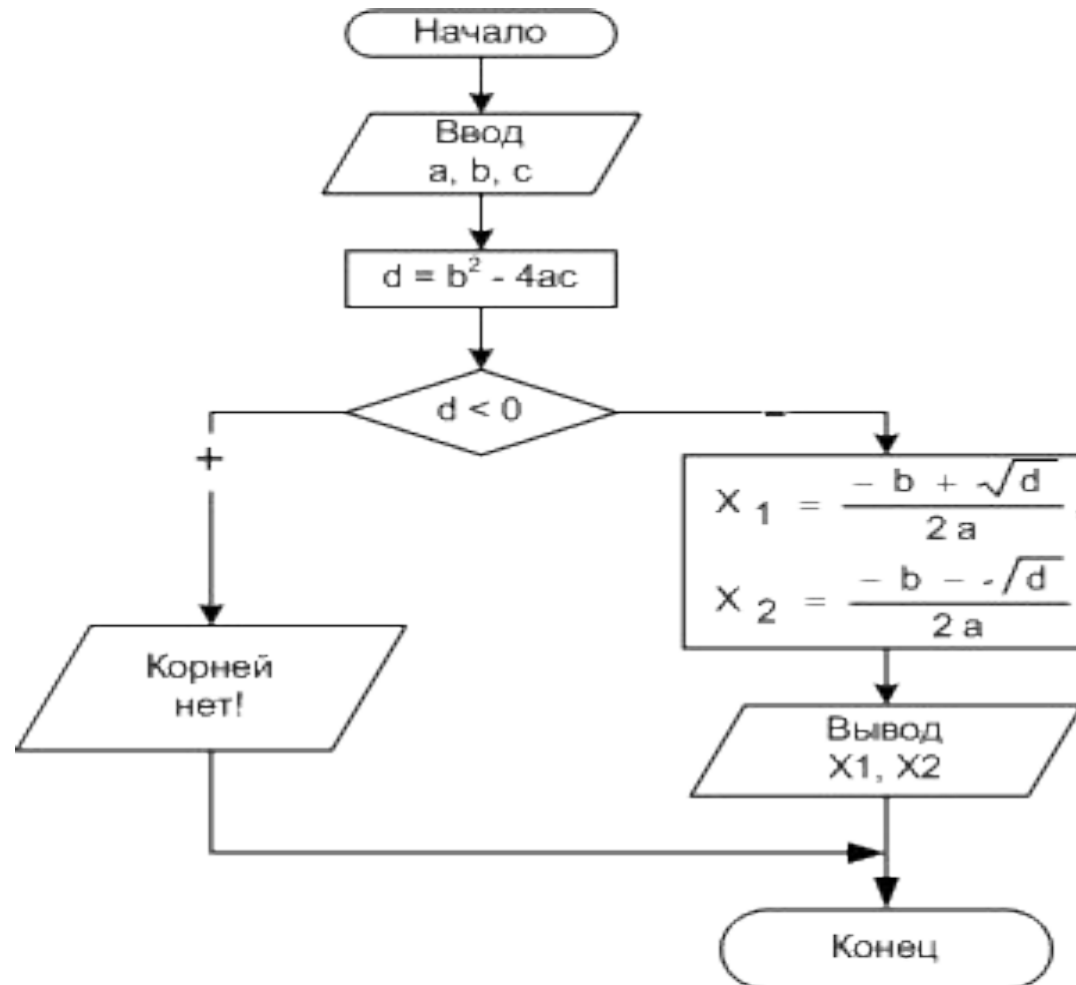


# Алгоритмы разветвленной структуры

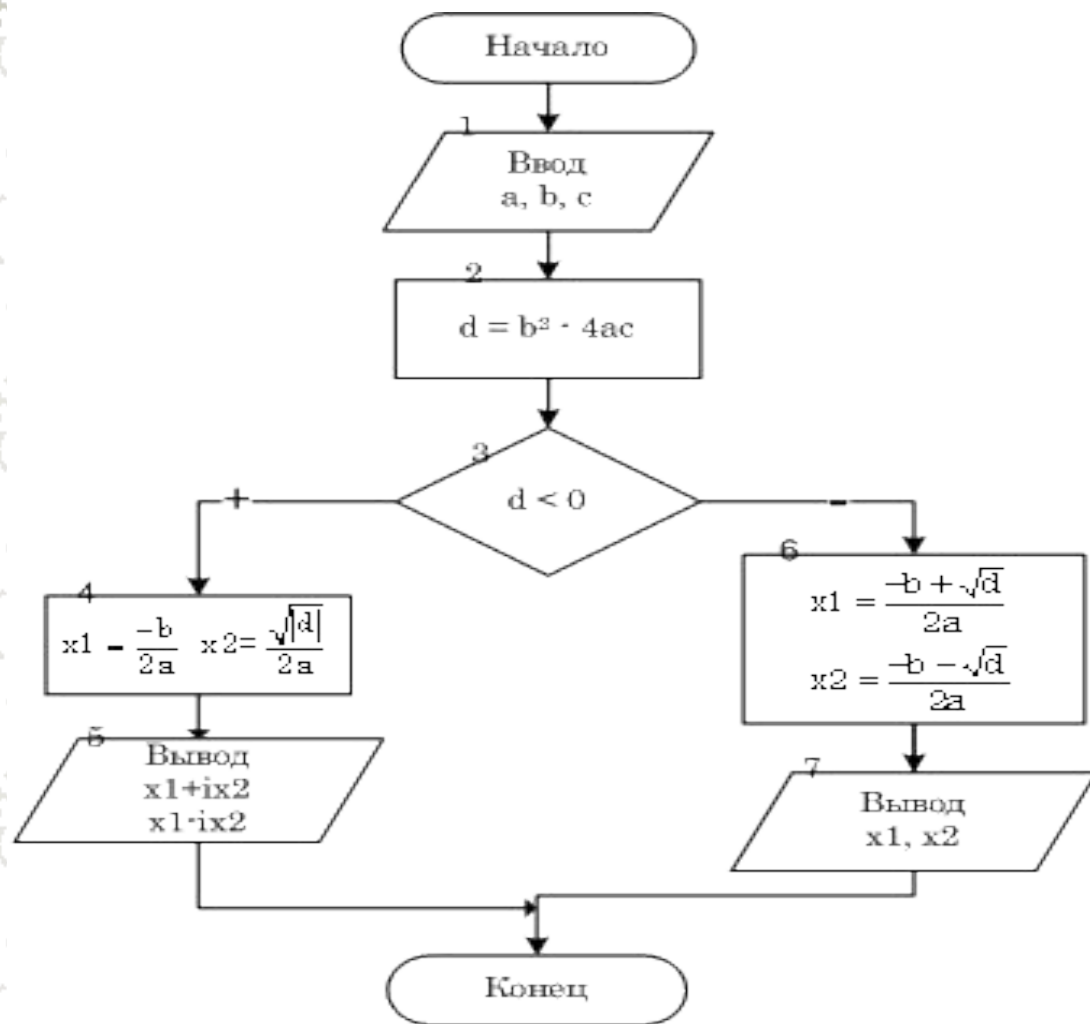




В качестве примера рассмотрим блок-схему алгоритма решения квадратного уравнения



# БЛОКЛОК-СХЕМА РЕШЕНИЯ КВАДРАТНОГО УРАВНЕНИЯ





# **Благодарю за внимание**

Презентацию подготовила преподаватель  
ГБОУ СПО «Баймакский  
сельскохозяйственный техникум»

**Мусина Ж.М.**

