

Основы алгоритмизации

Этапы разработки программ

- 1. Постановка задачи, разработка математической модели.**
- 2. Выбор метода численного решения.**
- 3. Построение алгоритма.**
- 4. Разработка программы.**
- 5. Отладка и испытание программы.**
- 6. Решение задачи на ЭВМ, обработка и оформление результатов расчета.**

Определение 1 **Алгоритмом** называется конечная последовательность строго выделенных правил, на основании исходных данных приводящих к однозначному решению задачи.

Определение 2. Под **алгоритмом** принято понимать «точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату».

Алгоритм должен обладать следующими свойствами:

- **Дискретность** (прерывность, раздельность) - алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов. Каждое действие, предусмотренное алгоритмом, исполняется только после того, как закончилось исполнение предыдущего.
- **Определенность** - каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.
- **Результативность** (конечность) - алгоритм должен приводить к решению задачи за конечное число шагов.
- **Массовость** - алгоритм решения задачи разрабатывается в общем виде, то есть, он должен быть применим для некоторого класса задач, различающихся только исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

Существует несколько способов для описания алгоритмов:

- **формульный;**
- **словесный,**
- **на естественном языке;**
- **графический, в виде так называемых блок-схем;**
- **на языке программирования.**

Текст кода, вводимого в ЭВМ, обрабатывается специальной программой - **транслятором**. Различают два вида трансляторов:

- **Компилятор** преобразует текст программы, написанной на языке высокого уровня в эквивалентную рабочую программу в машинных кодах, которая после полного преобразования запускается в работу (Pascal).
- **Интерпретатор** производит трансляцию каждого отдельного оператора программы и сразу же его выполняет (BASIC).

Интересующая программа (транслятор) должна находиться в оперативной памяти в течении всего времени выполнения программы пользователя. При интерпретации скорость выполнения программы существенно снижается, однако, при этом имеется возможность диалогового режима подготовки программы, представленной на языке высокого уровня.

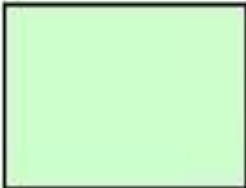
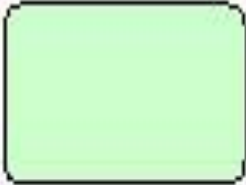
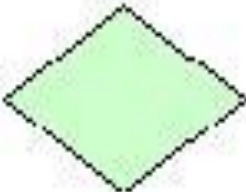
Виды ошибок

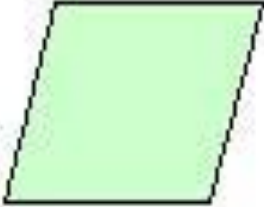
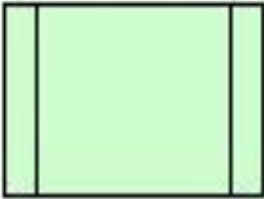
- Транслятор выявляет ошибки и сообщает о них, указывая их тип и место в программе. Такие ошибки называют **синтаксическими** ошибками.
- Ситуации, не разрешенные правилами языка, могут возникнуть и при выполнении программы, например деление на нуль или извлечение корня квадратного из отрицательного числа. Такие ошибки называют **ошибками времени выполнения**.
- Программа, не имеющая ошибок может и не дать верных результатов из-за так называемых логических ошибок в выбранном алгоритме, т.е **алгоритмических ошибок**.

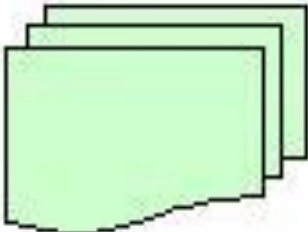
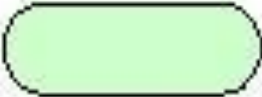
Блок-схема

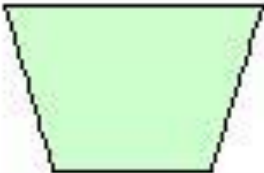
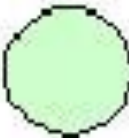
Блок-схема служит для изображения структур алгоритмов используя совокупность блочных символов (блоков), соединяемых линиями передач управления.

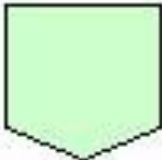
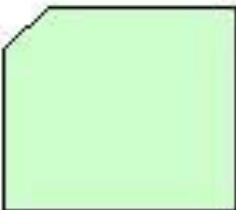
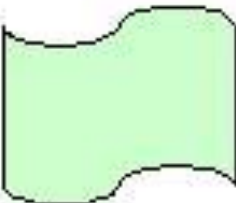
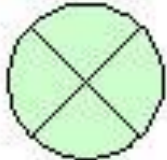
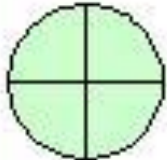
Условные графические обозначения схем алгоритмов и программ

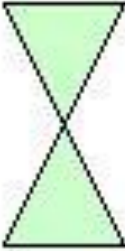
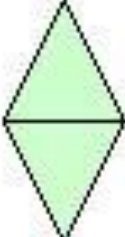
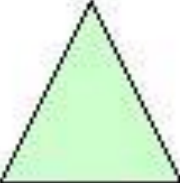
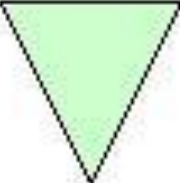
№ п/п	Наименование	Обозначение	Комментарий
1.	Процесс		Выполнение операции или группы операций, в результате которых изменяется значение, форма представления или расположение данных
2.	Альтернативный процесс		
3.	Решение		Выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий

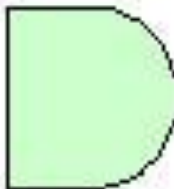
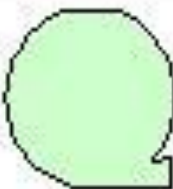
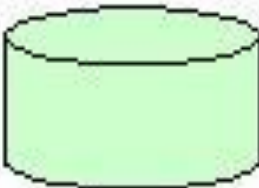
1.	Данные		Преобразование данных в форму, пригодную для обработки (ввод) или регистрация результатов обработки (вывод)
2.	Типовой процесс		Вычисления по подпрограмме, стандартной подпрограмме
3.	Внутренняя память		

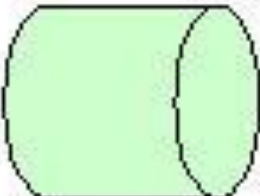
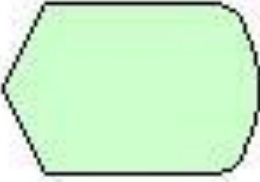
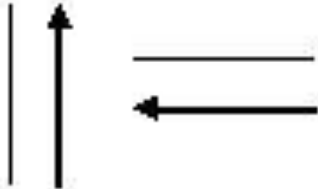
1.	Документ		
2.	Несколько документов		
3.	Знак завершения (пуск-останов)		Начало, конец, прерывание процесса обработки данных или выполнения программы
4.	Подготовка		Выполнение операций, меняющих команду или группу команд

1.	Ручной ввод		
2.	Ручное управление		
3.	Узел		

1.	Ссылка на другую страницу		
2.	Карточка		
3.	Перфолента		
4.	Узел суммирования		
5.	ИЛИ		

1.	Сопоставление		
2.	Сортировка		
3.	Извлечение		
4.	Объединение		

1.	Сохранённые данные		
2.	Задержка		
3.	Память с последовательным доступом		
4.	Магнитный диск		

1.	Память с прямым доступом		
2.	Дисплей		Ввод-вывод данных в случае, если непосредственно подключённое к процессору устройство воспроизводит данные и позволяет оператору вносить изменения в процессе их обработки
3.	Линии потока		Указание последовательности связей между символами
4.	Комментарий		Пояснения, содержание подпрограмм, формулы

Основные структуры алгоритмов

Основные структуры алгоритмов — это ограниченный набор блоков и стандартных способов их соединения для выполнения типичных последовательностей действий.

Структурный подход предполагает использование только нескольких основных структур, комбинация которых дает все многообразие алгоритмов и программ.

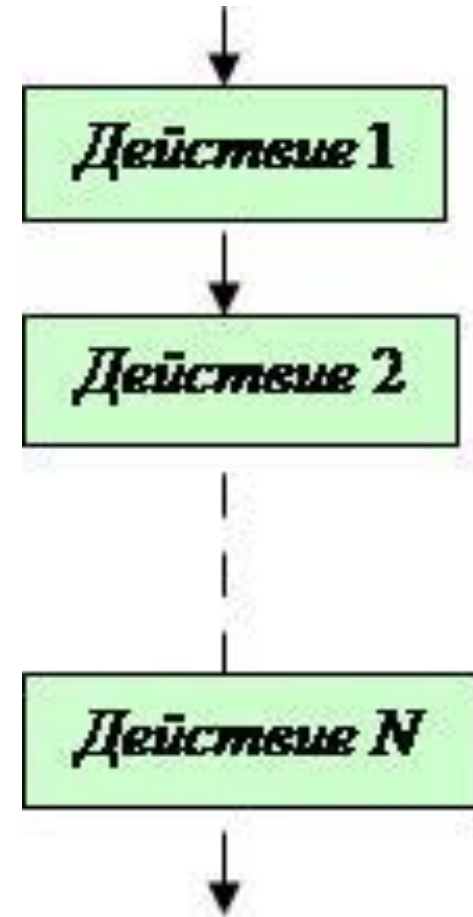
К основным структурам относятся:

- **следование;**
- **ветвление;**
- **повторение;**

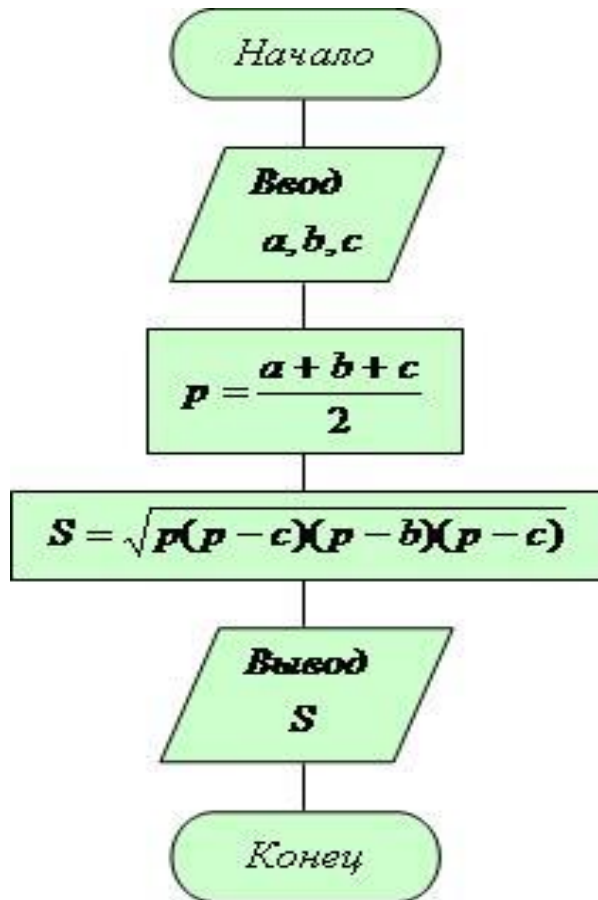
Каждая структура имеет единственный вход и единственный выход, причем составляющие их компоненты изображаются по вертикали. Исходя из структуры алгоритмов, различают алгоритмы линейного, разветвляющегося и циклического вычислительного процесса, а также алгоритмы со структурой вложенных циклов.

Алгоритмы линейной структуры

Блочные символы в этой структуре располагаются на схеме в том же порядке, в каком должны быть выполнены предписываемые ими действия. Такой порядок исполнения действий, называется естественным.



Составить схему алгоритма вычисления площади треугольника, полагая, что в качестве исходных данных заданы значения всех его сторон.



- **Листинг**

Алгоритмы разветвляющейся структуры

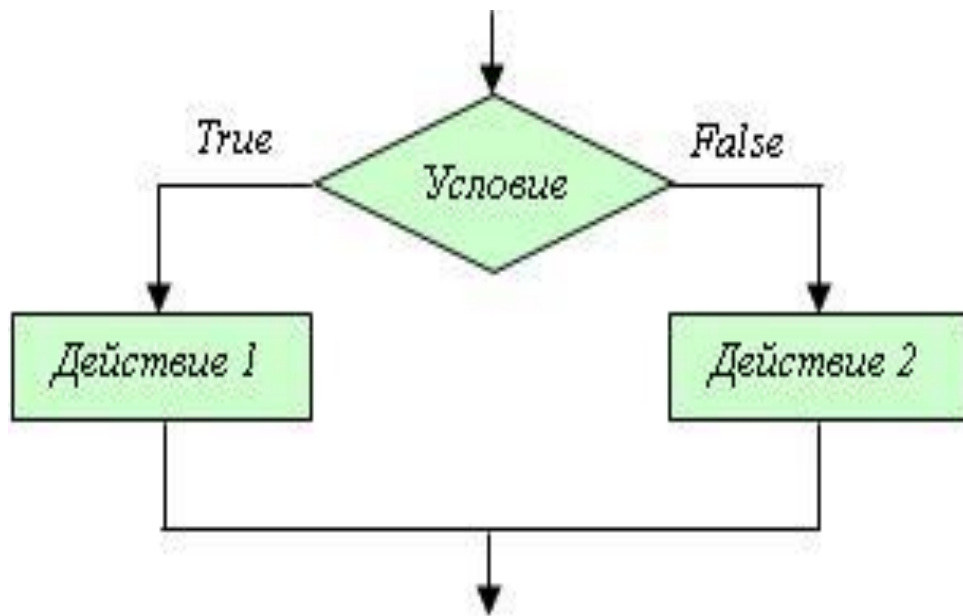
Разветвляющимся называют вычислительный процесс, в котором операции выполняются в одном из направлений в зависимости от заданных условий. В общем случае число ветвей в алгоритме разветвляющейся структуры не обязательно равно двум.

Отсюда, различают три вида структур разветвляющегося алгоритма.

Разветвление

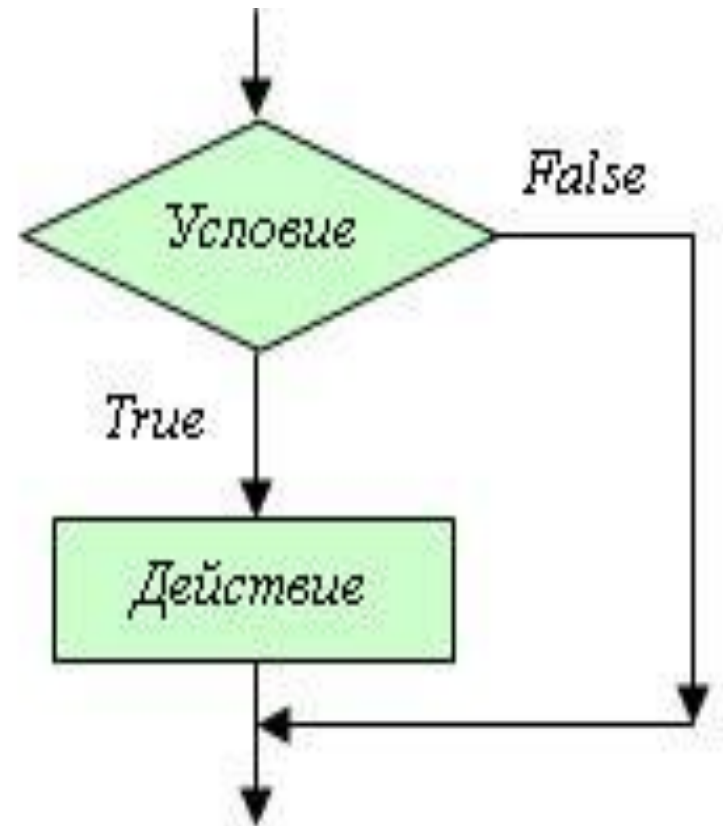
Применяется в том случае, когда в зависимости от условия нужно выполнить либо одно, либо другое действие.

Действие 1 или *Действие 2* может в свою очередь содержать несколько этапов

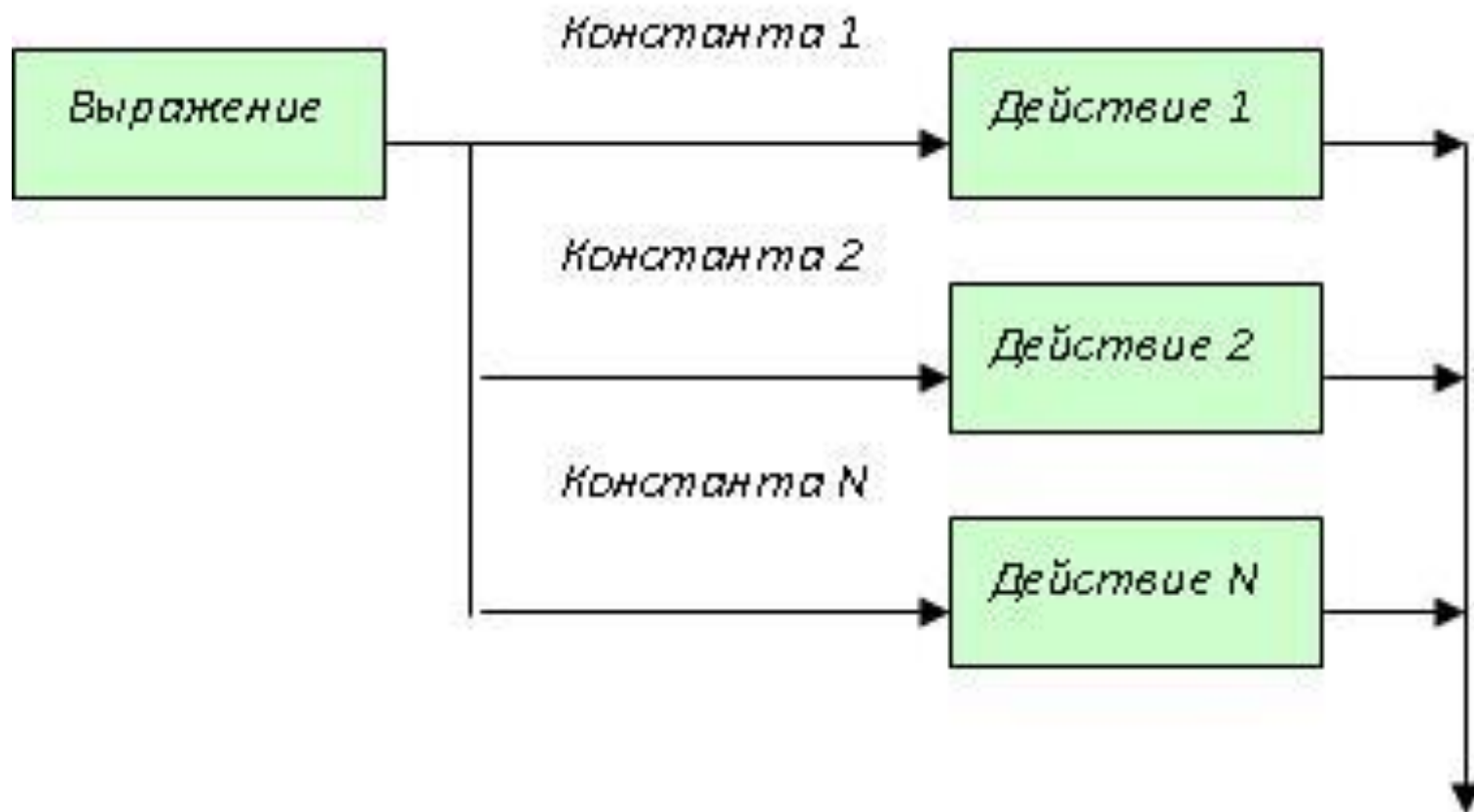


Обход

Частный случай
разветвления, когда
одна ветвь не
содержит никаких
действий



Множественный выбор



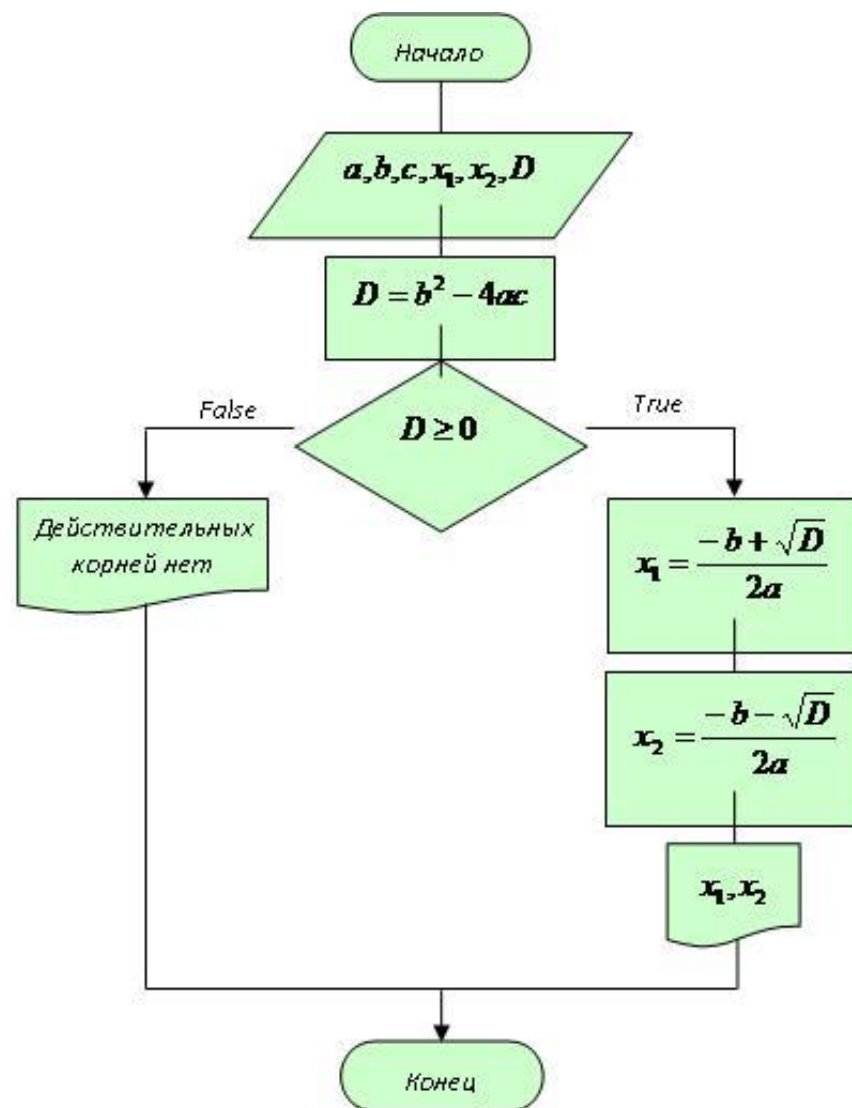
Составить схему алгоритма решения квадратного уравнения:

$$ax^2 + bx + c = 0$$

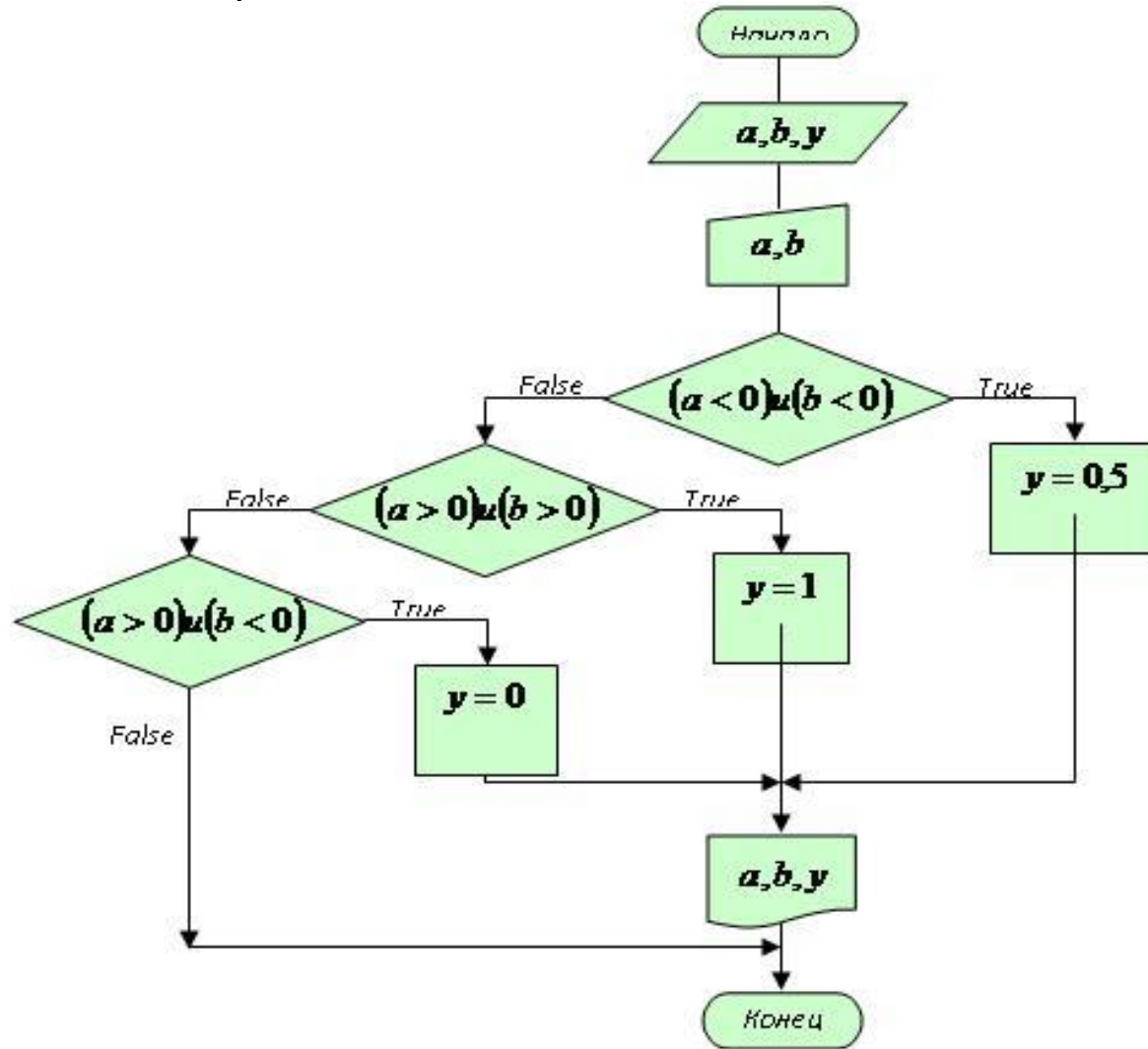
Математическая модель задачи:

В зависимости от значения подкоренного выражения, искомые результаты могут быть получены из выражений:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}; \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$



Если a и b отрицательные числа, присвоить Y значение 0,5;
если a и b положительные, присвоить Y значение 1;
если a положительное, а b отрицательное, присвоить Y значение 0.
Составить блок-схему вычисления значения .

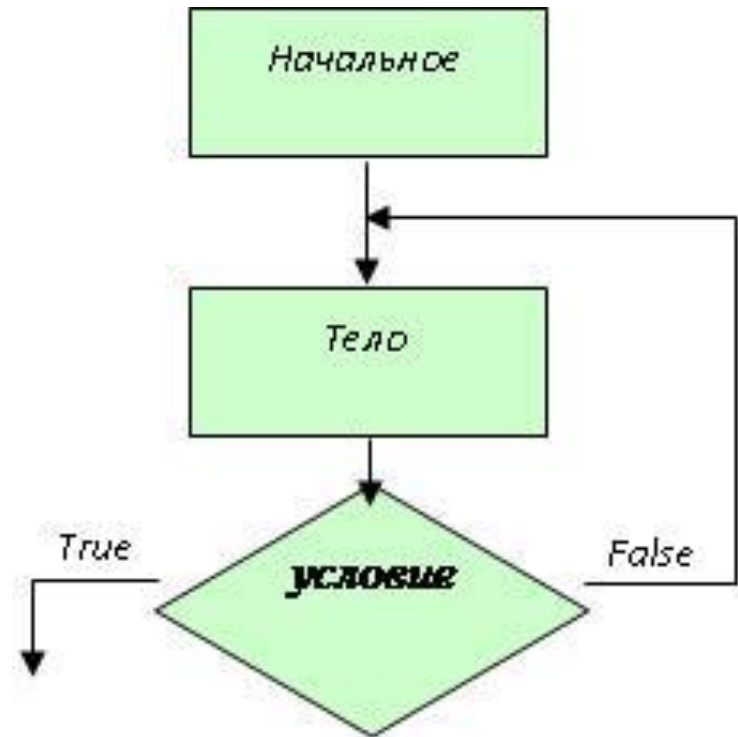


Алгоритмы циклической структуры

Цикл «До»

Применяется при необходимости выполнить какие-либо вычисления несколько раз до выполнения некоторого условия. Особенность этого цикла в том, что он всегда выполняется хотя бы один раз, так как первая проверка условия выхода из цикла происходит после того, как тело цикла выполнено.

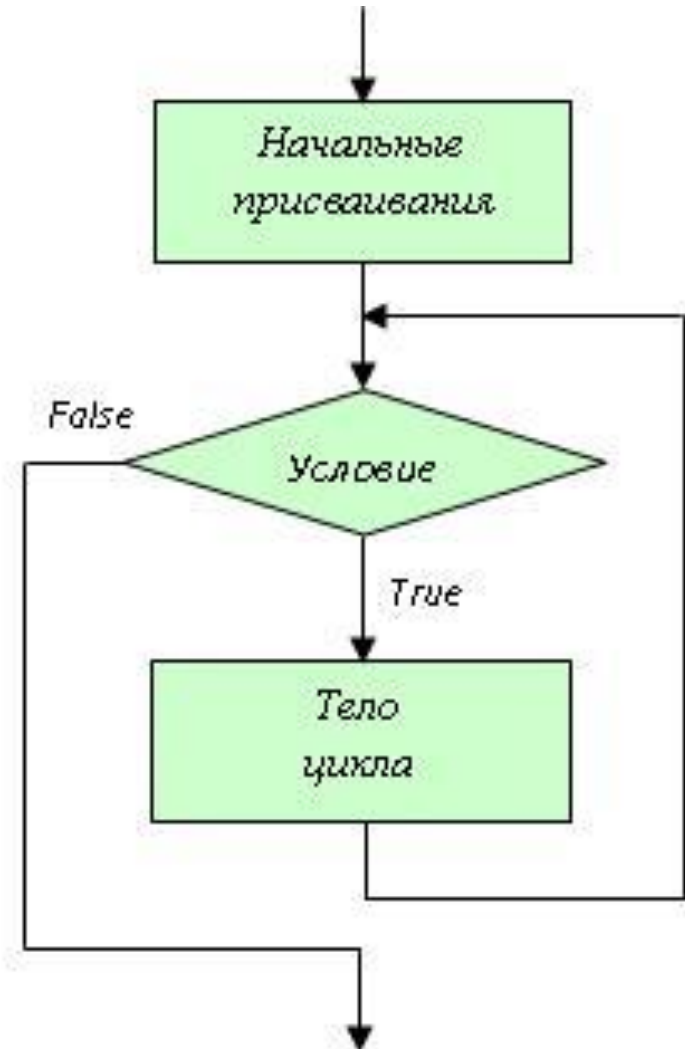
**Циклы Do с
постусловием**



Цикл «Пока»

Отличается от цикла *До* тем, что проверка условия проводится до выполнения тела цикла, и если при первой проверке условие выхода из цикла не выполняется ни разу.

Циклы *Do* с предусловием



Счетный цикл (цикл с управляющей переменной)

Цикл For

