

Алгоритмы и способы их описания

Понятие алгоритма

Алгоритм — это точное предписание, которое определяет процесс, ведущий от исходных данных к требуемому конечному результату.

Пример: правила сложения, умножения, решения алгебраических уравнений, умножения матриц и т.п.

К сведению: Слово алгоритм происходит от *algoritmi*, являющегося латинской транслитерацией арабского имени хорезмийского математика IX века **аль-Хорезми**. Благодаря латинскому переводу трактата **аль-Хорезми** европейцы в XII веке познакомились с позиционной системой счисления, и в средневековой Европе алгоритмом называлась десятичная позиционная система счисления и правила счета в ней.

Понятие алгоритма

Применительно к ЭВМ алгоритм определяет вычислительный процесс, начинающийся с обработки некоторой совокупности возможных исходных данных и направленный на получение определенных этими исходными данными результатов. Термин **вычислительный процесс** распространяется и на обработку других видов информации, например, символьной, графической или звуковой.

Основные свойства алгоритмов

1. **Результативность** означает возможность получения результата после выполнения конечного количества операций.
2. **Определенность** состоит в совпадении получаемых результатов независимо от пользователя и применяемых технических средств.
3. **Массовость** заключается в возможности применения алгоритма к целому классу однотипных задач, различающихся конкретными значениями исходных данных.
4. **Дискретность** — возможность расчленения процесса вычислений, предписанных алгоритмом, на отдельные этапы, возможность выделения участков программы с определенной структурой.

Задание алгоритма

Для задания алгоритма необходимо описать следующие его элементы:

- ⦿ набор объектов, составляющих совокупность возможных исходных данных, промежуточных и конечных результатов;
- ⦿ правило начала;
- ⦿ правило непосредственной переработки информации (описание последовательности действий);
- ⦿ правило окончания;
- ⦿ правило извлечения результатов.

СПОСОБЫ ОПИСАНИЯ алгоритмов

- Словесно - формульный;
- структурный или блок - схемный;
- с помощью графов - схем;
- с помощью сетей Петри.

Словесно – формульный алгоритм

При словесно-формульном способе алгоритм записывается в виде текста с формулами по пунктам, определяющим последовательность действий.

Пример: необходимо найти значение следующего выражения: $y = 2a - (x+6)$.

Словесно-формульным способом алгоритм решения этой задачи может быть записан в следующем виде:

1. Ввести значения a и x .
2. Сложить x и 6 .
3. Умножить a на 2 .
4. Вычесть из $2a$ сумму $(x+6)$.
5. Вывести y как результат вычисления выражения.

Блок - схемы

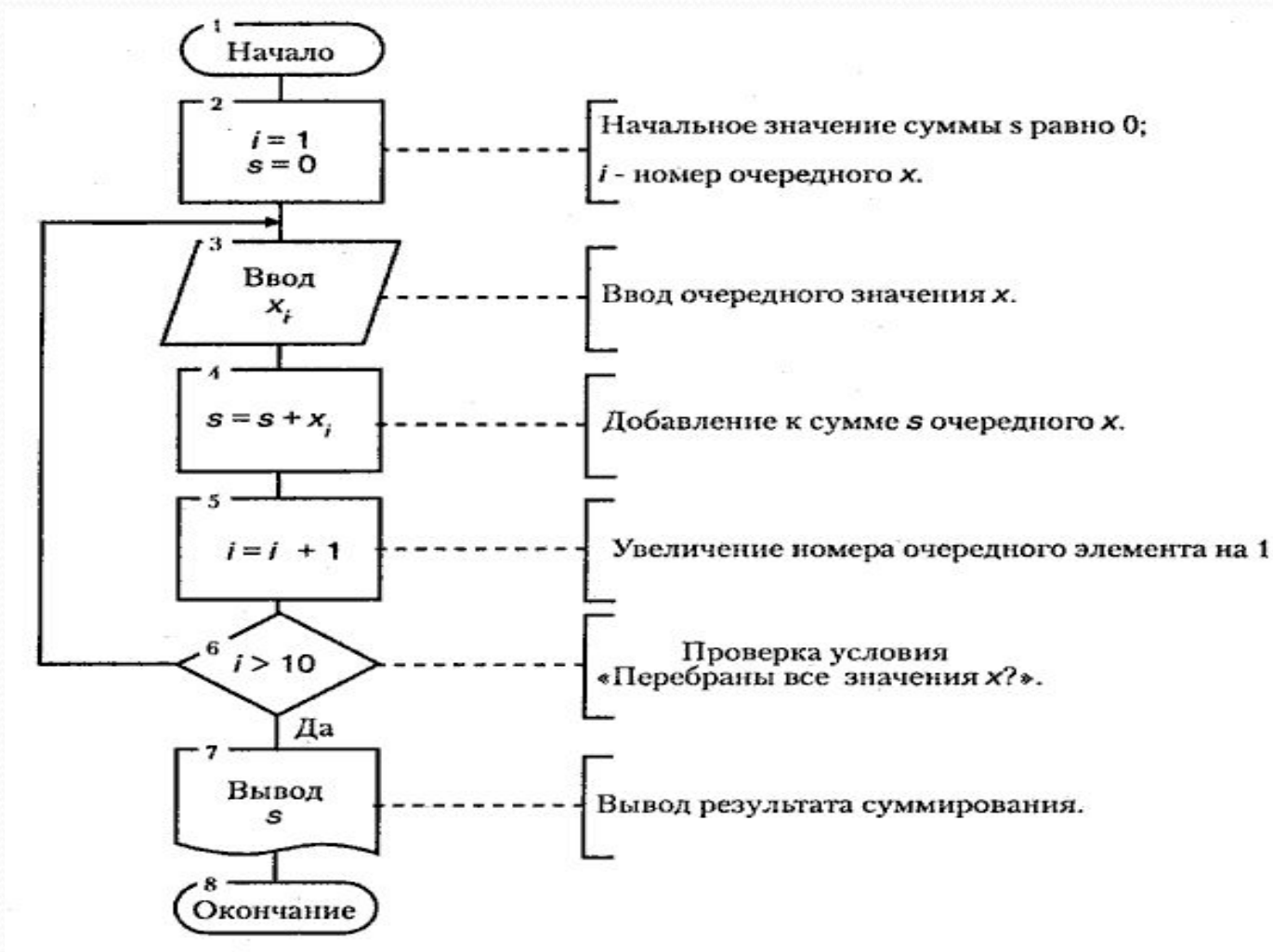
При блок - схемном описании алгоритм изображается геометрическими фигурами (блоками), связанными по управлению линиями (направлениями потока) со стрелками. В блоках записывается последовательность действий.

Преимущества:

- 1.наглядность: каждая операция вычислительного процесса изображается отдельной геометрической фигурой.
- 2.графическое изображение алгоритма наглядно показывает разветвления путей решения задачи в зависимости от различных условий, повторение отдельных этапов вычислительного процесса и Другие детали.

К сведению: Оформление программ должно соответствовать определенным требованиям. В настоящее время действует единая система программной документации (ЕСПД), которая устанавливает правила разработки, оформления программ и программной документации. В ЕСПД определены и правила оформления блок-схем алгоритмов (ГОСТ 10.002-80 ЕСПД, ГОСТ 10.003-80 ЕСПД).

Пример блок - схемы








Алгоритм нахождения суммы 10-ти чисел

Блоки на блок - схемах

Операции обработки данных и носители информации изображаются на схеме соответствующими блоками.

Большая часть блоков по построению условно вписана в прямоугольник со сторонами a и b . Минимальное значение $a = 10$ мм, увеличение a производится на число, кратное 5 мм. Размер $b = 1,5a$. Для отдельных блоков допускается соотношение между a и b , равное $1:2$. В пределах одной схемы рекомендуется изображать блоки одинаковых размеров. Все блоки нумеруются.

Виды блоков

Наименование	Обозначение	Функции
Процесс		Выполнение операции или группы операций, в результате которых изменяется значение, форма представления или расположение данных.
Ввод-вывод		Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод).
Решение		Выбор направления выполнения алгоритма в зависимости от некоторых переменных условий.
Предопределенный процесс		Использование ранее созданных и отдельно написанных программ (подпрограмм).
Документ		Вывод данных на бумажный носитель.

Правила создания блок - схем

1. Линии, соединяющие блоки и указывающие последовательность связей между ними, должны проводится параллельно линиям рамки.
2. Стрелка в конце линии может не ставиться, если линия направлена слева направо или сверху вниз.
3. В блок может входить несколько линий, то есть блок может являться преемником любого числа блоков.
4. Из блока (кроме логического) может выходить только одна линия.
5. Логический блок может иметь в качестве продолжения один из двух блоков, и из него выходят две линии.
6. Если на схеме имеет место слияние линий, то место пересечения выделяется точкой. В случае, когда одна линия подходит к другой и слияние их явно выражено, точку можно не ставить.
7. **Схему алгоритма** следует выполнять как единое целое, однако в случае необходимости допускается обрывать линии, соединяющие блоки.

Структурные схемы алгоритмов

- Последовательность двух или более операций;
- выбор направления;
- повторение.

Любой вычислительный процесс может быть представлен как комбинация этих элементарных алгоритмических структур.

Виды алгоритмов

- линейные;
- ветвящиеся;
- циклические.

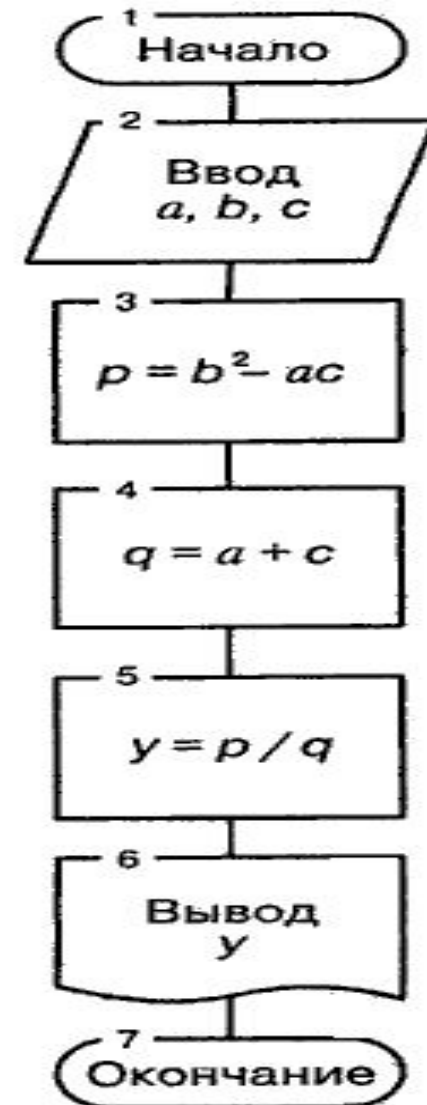
Линейные алгоритмы

В линейном алгоритме операции выполняются последовательно, в порядке их записи. Каждая операция является самостоятельной, независимой от каких-либо условий. На схеме блоки, отображающие эти операции, располагаются в линейной последовательности.

Линейные алгоритмы имеют место, например, при вычислении арифметических выражений, когда имеются конкретные числовые данные и над ними выполняются соответствующие условию задачи действия.

Пример линейного алгоритма

Составить блок – схему алгоритма вычисления арифметического выражения $y = (b^2 - ac) : (a + c)$



Алгоритм с ветвлением

Алгоритм называется **ветвящимся**, если для его реализации предусмотрено несколько направлений (ветвей). Каждое отдельное направление алгоритма обработки данных является отдельной ветвью вычислений.

Ветвление в программе — это выбор одной из нескольких последовательностей команд при выполнении программы. Выбор направления зависит от заранее определенного признака, который может относиться к исходным данным, к промежуточным или конечным результатам. Признак характеризует свойство данных и имеет два или более значений.

Ветвящийся процесс, включающий в себя две ветви, называется **простым**, более двух ветвей — **сложным**.

Сложный ветвящийся процесс можно представить с помощью простых ветвящихся процессов.

Алгоритм с ветвлением

Направление ветвления выбирается логической проверкой, в результате которой возможны два ответа:

1. «да» — условие выполнено
2. «нет» — условие не выполнено.

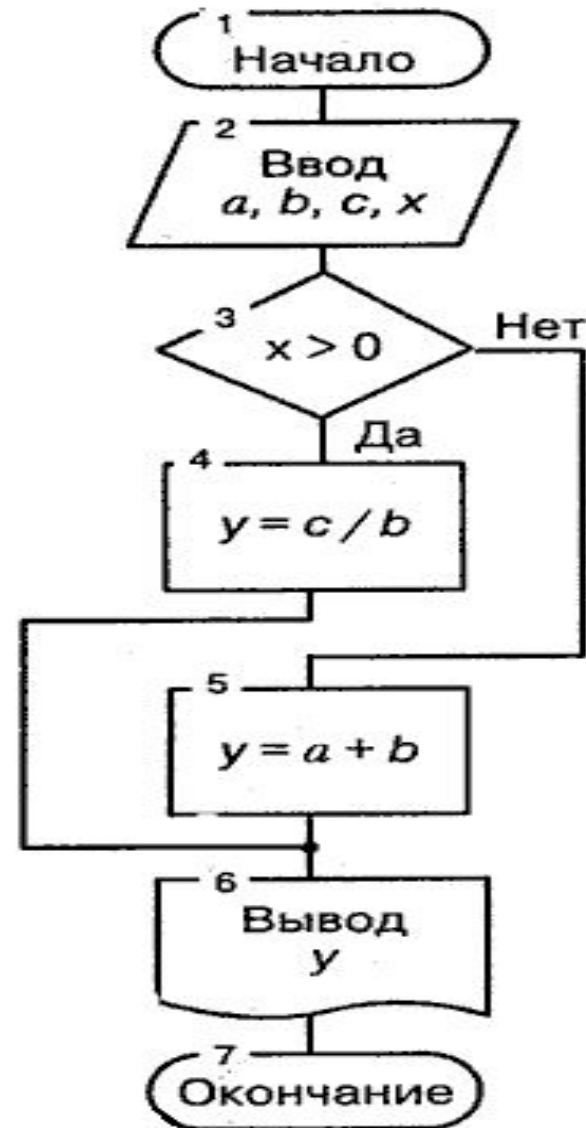
Следует иметь в виду, что, хотя на схеме алгоритма должны быть показаны все возможные направления вычислений в зависимости от выполнения определенного условия (или условий), при однократном прохождении программы процесс реализуется только по одной ветви, а остальные исключаются.

Важно! Любая ветвь, по которой осуществляются вычисления, должна приводить к завершению вычислительного процесса.

Пример алгоритма с ветвлением

Составить блок-схему алгоритма с ветвлением для вычисления следующего выражения:

$$Y = (a+b), \text{ если } X < 0;$$
$$c/b, \text{ если } X > 0.$$



Циклические алгоритмы

Циклическими называются алгоритмы, содержащие циклы.

Цикл — это многократно повторяемый участок алгоритма.

Этапы организации цикла

- подготовка (инициализация) цикла (**И**);
- выполнение вычислений цикла (тело цикла) (**Т**);
- модификация параметров (**М**);
- проверка условия окончания цикла (**У**).

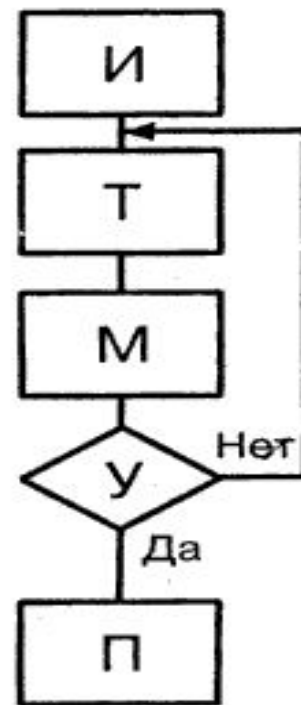
Порядок выполнения этих этапов, например, **Т** и **М**, может изменяться.

Типы циклов

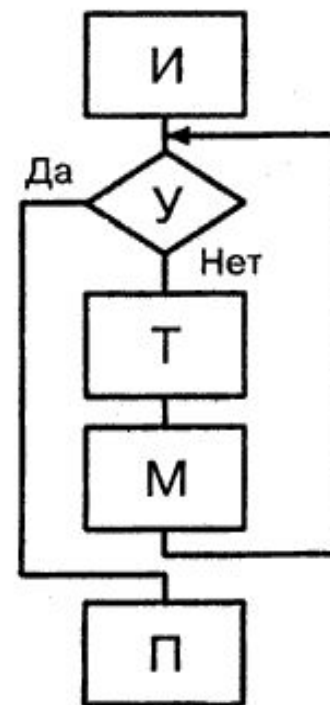
В зависимости от расположения проверки условия окончания цикла различают циклы с нижним и верхним окончаниями.

Для цикла с нижним окончанием (рис. а) тело цикла выполняется как минимум один раз, так как сначала производятся вычисления, а затем проверяется условие выхода из цикла.

В случае цикла с верхним окончанием (рис. б) тело цикла может не выполниться ни разу в случае, если сразу соблюдается условие выхода.



а



б

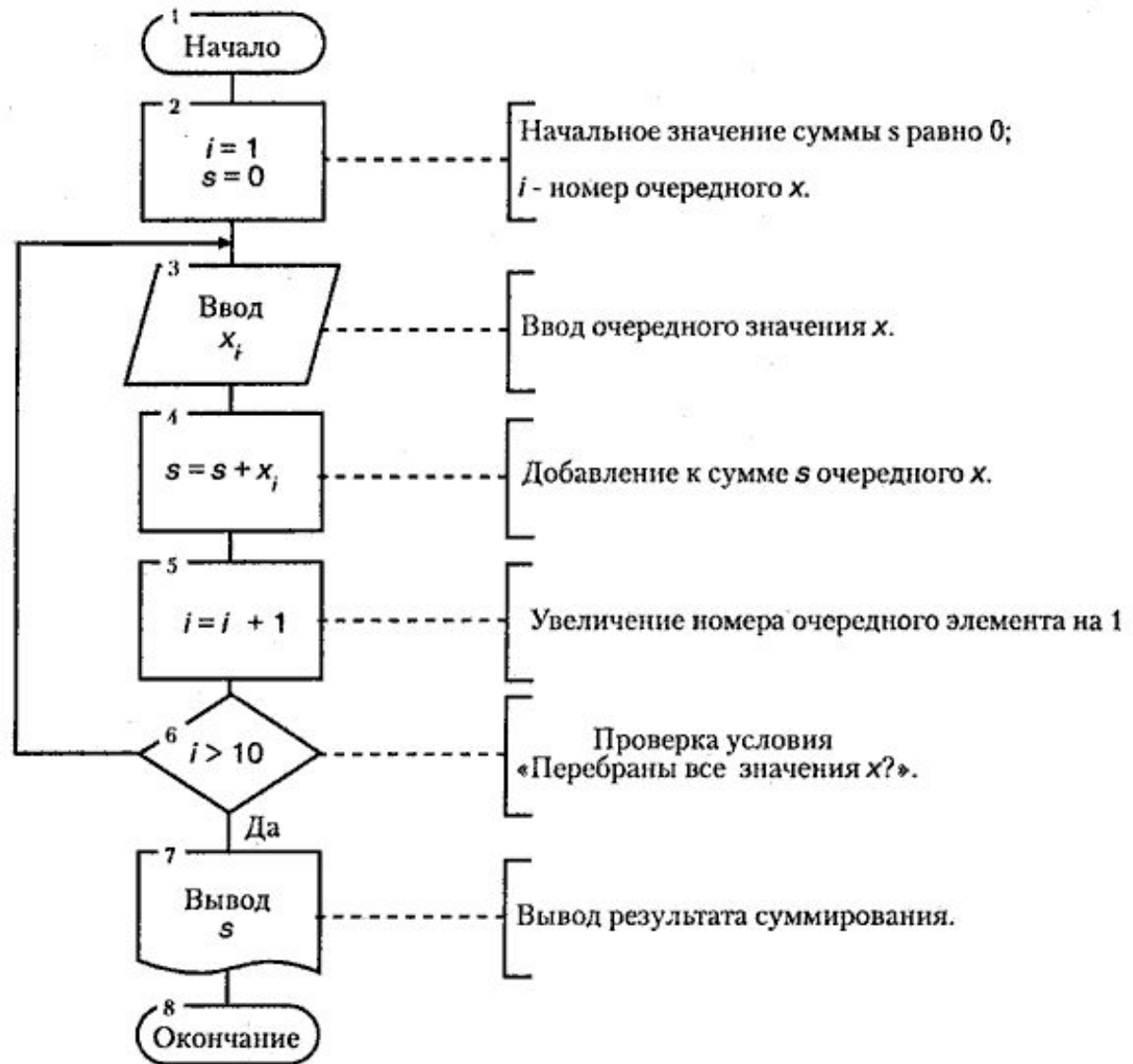
Примеры циклических алгоритмов

Виды циклов

- Цикл называется детерминированным, если число повторений тела цикла заранее известно или определено.
- Цикл называется итерационным, если число повторений тела цикла заранее неизвестно, а зависит от значений параметров (некоторых переменных), участвующих в вычислениях.

Пример циклического алгоритма

Алгоритм нахождения суммы 10-ти чисел



Этапы подготовки и решения задач на ЭВМ

На ЭВМ могут решаться задачи различного характера, например:

научно-инженерные; разработки системного программного обеспечения; обучения; управления производственными процессами и т. д.

В процессе подготовки и решения на ЭВМ научно -инженерных задач можно выделить следующие **этапы**:

1. постановка задачи;
2. математическое описание задачи;
3. выбор и обоснование метода решения;
4. алгоритмизация вычислительного процесса;
5. составление программы;
6. отладка программы;
7. решение задачи на ЭВМ и анализ результатов.

В задачах другого класса некоторые этапы могут отсутствовать, например, в задачах разработки системного программного обеспечения отсутствует математическое описание.