Основы алгоритмизации

Этапы решения задачи. Под решением конкретной задачи понимают не только определение результатов с помощью ЭВМ, а также всю подготовительную работу, которую необходимо выполнить для достижения поставленной цели. Поэтому весь процесс можно разбить на несколько этапов:

- постановка задачи;
- формализация (математическая постановка задачи);
- выбор метода решения;
- алгоритмизация задачи;
- программирование;
- отладка программы;
- решение задачи на ЭВМ и анализ результатов.

Применительно к решению задач на ЭВМ алгоритм представляет последовательность арифметических и логических действий над числовыми значениями переменных, приводящих к вычислению решения задачи при изменении исходных данных в довольно широких пределах. Каждый алгоритм разбивает весь вычислительный процесс на отдельные этапы и содержит информацию как о тех действиях, которые надо выполнить на любом из этапов, так и о порядке, выполнения этих этапов. По алгоритму складывается программа. Процесс создания программ имеет название программирования. Программа ЭВМ - это описание алгоритма решения задачи с помощью алгоритмического языка. В ЭВМ она представленна набором машинных инструкций, с помощью которых закодирован алгоритм решения задачи или управление процессом.

Способы описания алгоритмов. В процессе разработки алгоритма могут использоваться разные способы его описания, отличающиеся простотой, наглядностью, степенью формализации. В практике

программирования применяются следующие способы: словесное описание алгоритма; описание алгоритма в виде формул; словесно-формульное; табличная форма описания (используется для ручного счета и в пакетах); описание алгоритма в виде блок-схем (схем алгоритмов); операторный способ описания алгоритма; описание алгоритма алгоритмическим языком.

Наибольшее распространен способ описания алгоритма в виде блок-схемы, который представляет собой графическую интерпретацию логической схемы решения задач. Блок-схемой - называется графическое изображение алгоритма, когда отдельные действия изображаются различными геометрическими фигурами – блоками. Правила выполнение схем алгоритмов регламентирует ГОСТ 19.002-80, графические символы, которые используются, регламентирует ГОСТ 19.003-80 (табл. 1). Графические символы должны иметь порядковые номера, которые проставляются в левой части верхней стороны изображения в разрыве линии контура. Графические символы на схеме алгоритма соединяются линиями потока информации. Основное направление потока идет сверху вниз и слева направо (стрелки направления на линиях потока могут не указываться). В других случаях применение стрелок обязательно. Количество входящих линий не ограничено. Выходящая линия может быть лишь одна (исключение - блок проверки логических условий и блок модификации). Линию потока информации подводят, как правило, к середине графического символа.

В блоках приняты размеры:

A= 10, 15, 20,.... мм;

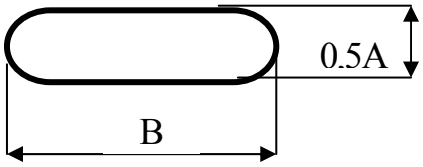
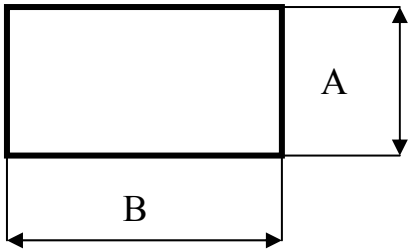
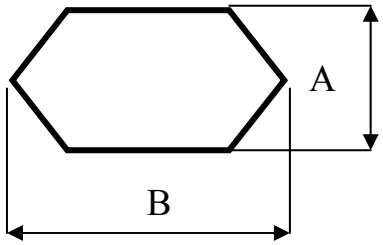
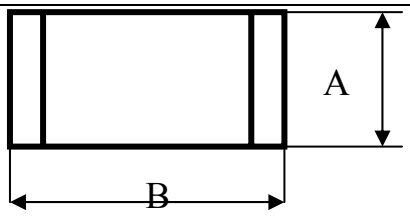
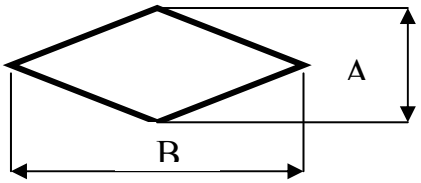
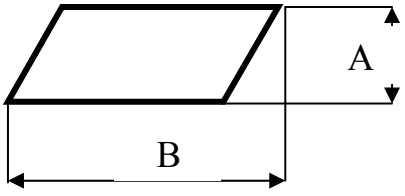
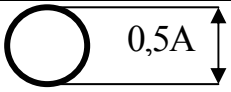
B=1,5A (допускается устанавливать B=2A).

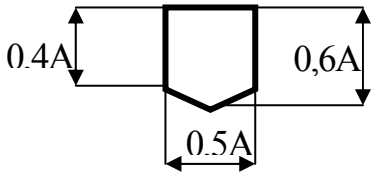
При необходимости увеличения размеров схемы алгоритма допускается увеличение размера A на число, кратное 5.

При выполнении схем алгоритмов необходимо видерживать минимальное расстояние между параллельными линиями потока информации 3 мм и 5 мм – между другими символами.

Таблица 1

Условные графические обозначения,
применяемые при составление схем алгоритмов

Название блока	Графическое обозначение блока	Выполняемые действия
Пуск-останов		Начало, конец, преры-вання процесса обработки или выполнения программы
Процесс		Выполнение операций, в результате которой изменяется значение, форма представления или расположение данных
Модификация (заголовок цикла)		Выполнение операций, меняющих команды, или группы команд, изме-няющих программу
предопределен-ный процесс (подпрограмма)		Использование прежде созданных или отдельно написанных алгоритмов или программ
Решение		Выбор направления виполнения алгоритма или программы в зависимости от некоторых переменных условий
Ввод- вывод		Преобразование данных в форму, пригодную для обработки (ввод), или отображение результатов обработки (вывод)
Соединитель страничный		Разрыв линии потока информации

Соединитель межстраничный		Разрыв линии потока между страницами

Линейные алгоритмы. При составление схем алгоритмов необходимо отличать линейные, разветвленные и циклические алгоритмы. Как правило, они не используются в чистом виде и обычно схема алгоритма достаточно сложной задачи представляет собой композицию перечисленных видов алгоритмов.

Линейным называется вычислительный процесс, в котором действия выполняются последовательно в естественном и единственном порядке следования. Блочные символы в этой структуре размещаются в том же порядке, в котором должны быть выполнены предложеные действия.

В алгоритме линейной структуры используются следующие блочные символы:

- пуск (начало);
- ввод;
- процесс;
- вывод;
- останов (конец).

Пример: вычислить высоты треугольника с сторонами a, b, c, используя формулы

$$h_a = \frac{2}{a}\sqrt{p(p-a)(p-b)(p-c)},$$

$$h_b = \frac{2}{b}\sqrt{p(p-a)(p-b)(p-c)},$$

$$h_c = \frac{2}{c}\sqrt{p(p-a)(p-b)(p-c)},$$

где $p = \frac{a+b+c}{2}$.

Чтобы исключить повторяющиеся числа, используем промежуточную величину

$$t = 2\sqrt{(p-a)(p-b)(p-c)}\ ,$$

тогда $h_a = t/a, \qquad h_b = t/b, \qquad h_c = t/c.$

Значение величин p, t, h_a, h_b, h_c сохраняются в ячейках памяти с соответствующими именами. Алгоритм вычисления представлен на рис. 1.

О

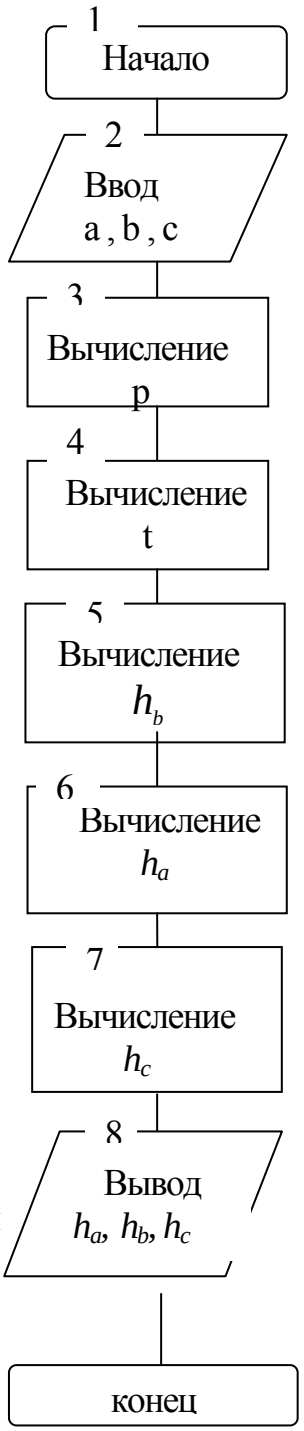


Рис. 1 Алгоритм линейной структуры

Разветвленные алгоритмы. На практике часто возникает необходимость, в зависимости от полученных исходных данных или значений промежуточных результатов, осуществлять вычисление по одним или другим формулам, то есть в зависимости от выполнения какого-нибудь логического условия вычислительный процесс должен идти по одному или другому направлению. Алгоритмы, содержащие действие выбора направления вычислительного процесса, имеют название разветвленных. Разветвление на блок-схемах представляется логическим блоком выбора. Условие разветвления записывается внутри блока логическим отношением или логическим выражением.

Логическое отношение - последовательная запись констант, переменных, арифметических выражений, объединенных операциями отношение .(>=;>;=;<>;<;<=)

Логическое выражение - последовательная запись логических отношений, разделенных знаками логических операций:

- логического умножения или операции "И" (AND);
- логического добавления или операции "ИЛИ" (OR);
- логического возражения или операции "НЕ" (NOT).

Пример: Вичислить корни квадратного уравнения

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

$x_{1,2} = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$, если $D = b^2 - 4ac \geq 0$

$x_{1,2} = \alpha \pm i\beta$, если $D = b^2 - 4ac \leq 0$

Блок-схема алгоритма представленна на рис. 2.

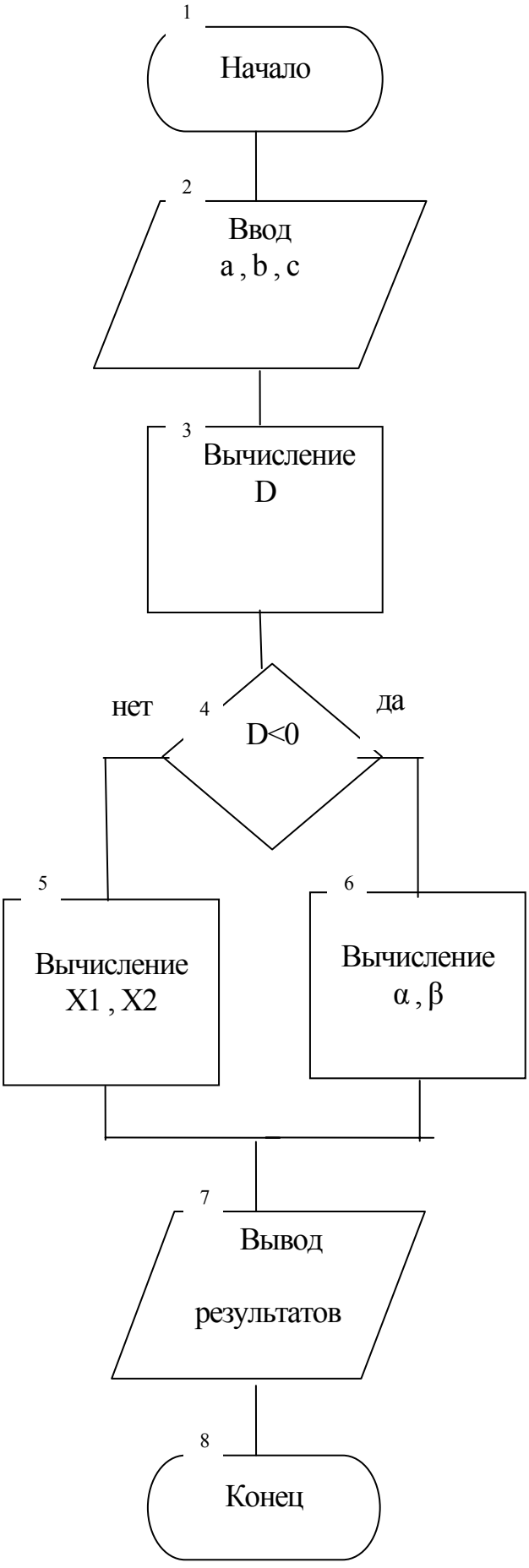


Рис. 2 Алгоритм разветвленной структуры

Алгоритмы циклической структуры. Решение многих задач сводится к выполнению вычислений по одним и тем же математическим зависимостям, но при разных значениях входящих величин. Такой вычислительный процесс называется циклическим, а многократно повторяющиеся участки этого процесса называются циклами. Переменные, изменяемые при каждом новом выходе на повторение, называют параметрами цикла. Переменная, значение которой вычисляется и сохраняется в одной и той же ячейке памяти ЭВМ, называется простой переменной. Переменная, являющаяся элементом массива, называется переменной с индексом. При использовании простой переменной, она является параметром цикла. При использовании переменной с индексом параметром цикла является ее индекс. В одном цикле может быть несколько параметров.

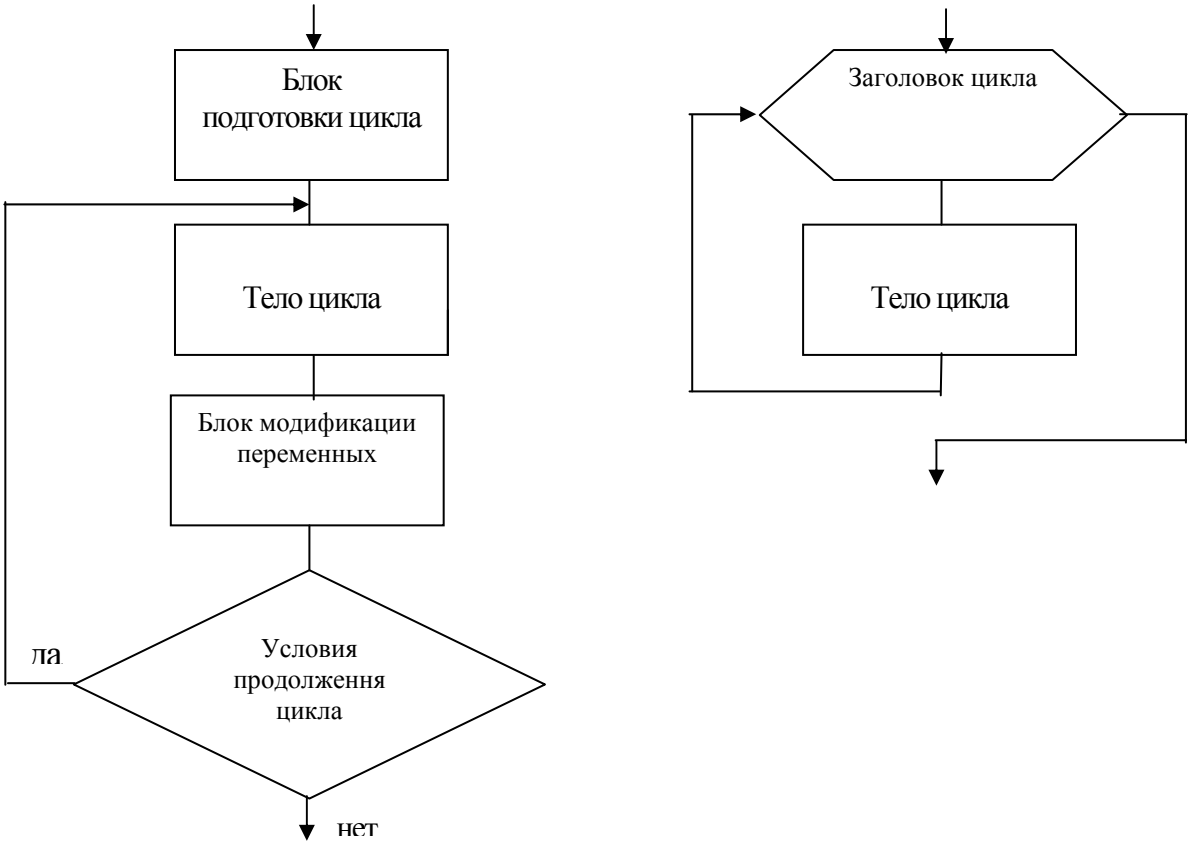


Рис. 3 Организация циклических структур

Арифметические циклы. Различают регулярные или арифметические циклы (с известным числом повторений), условием окончания которых является достижение параметром цикла своего конечного значения, и циклы итерационные (с неизвестным числом повторений). В таких циклах условие повторения или окончание цикла задается по некоторому промежуточному или окончательному результату, например, пока не будет достигнута необходимая точность вычислений.

Регулярные циклы называют также циклами с счетчиком. Число повторений тела цикла в этом случае подсчитывается с помощью введения специальной переменной - счетчика, для которой известны начальное, конечное значение и шаг ее изменения. Управление циклом осуществляется на основании сравнения текущего значения счетчика с заданным конечным. Число повторений тела цикла определяется по формуле:

$$N =](Xк - Xн)/h[+ 1,$$

где Xк - конечное значение параметра цикла;

Xн - начальное значение параметра цикла;

h - шаг изменения параметра цикла.

Текущее значение параметра цикла вычисляется по формуле:

$$X = Xн + (k - 1)h,$$

где k изменяется от 1 к N.

При реализации циклических вычислительных процессов используются рекурсивные выражения, которые описывают любой член последовательности чисел.

Например, формула X = X + 1, в которой реализована рекурсия, означает, что к содержимому ячейки памяти с именем X прибавляется 1 и результат записывается в X. Такая формула называется рекурентной и связывает между собой последовательность вычисления значений переменной X. Исходными данными для каждого последующего шага являются результаты предыдущего.

Пример организации простого цикла с использованием логического блока выбора и блока модификации: составить алгоритм вычисления выражения

$$Y = X^2 - SIN^2(X),$$

где X изменяется от Xн = 0 к Xк = 5 с шагом Hx = 0,15.

Схема алгоритма задачи (рис. 4) представляет простой циклический вычислительный процесс с неявно заданным числом повторений.

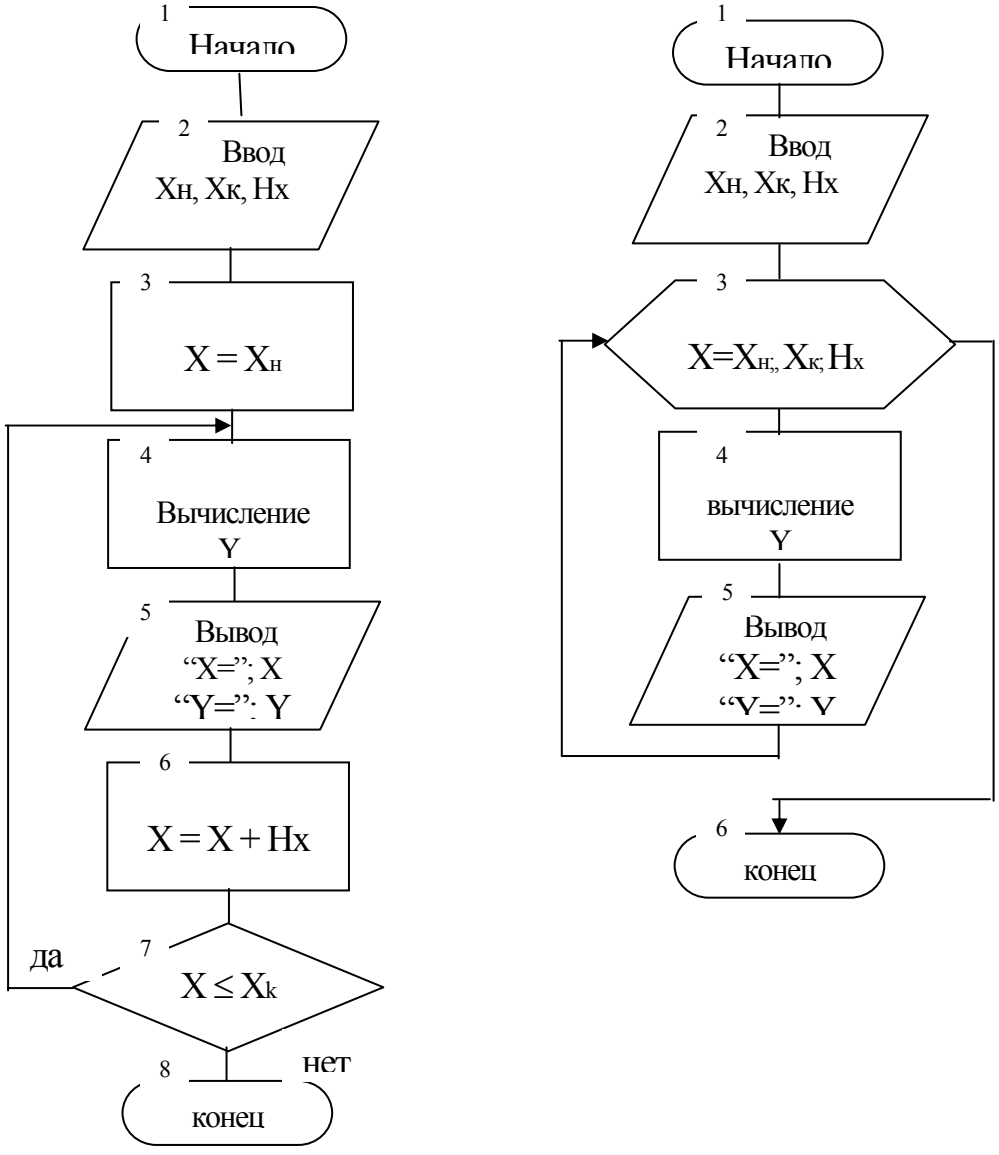


Рис. 4 Алгоритм простого циклического процесса

Понятие рекурсии используется также при вычислении суммы или произведения конечного количества чисел. При этом необходимо выполнить следующие действия:

- сформировать исходные данные;

- определить начальное состояние ячейки, в которой производится накопление суммы или произведения;

- организовать цикл накопления суммы или произведения;

- вывести результат.

Пример: составить алгоритм вычисления значение факториала (произведения чисел натурального ряда от 1 к N: 2!= 1*2; 3!= 1*2*3; 4!=1*2*3*4; N!=1*2*3*...*N).

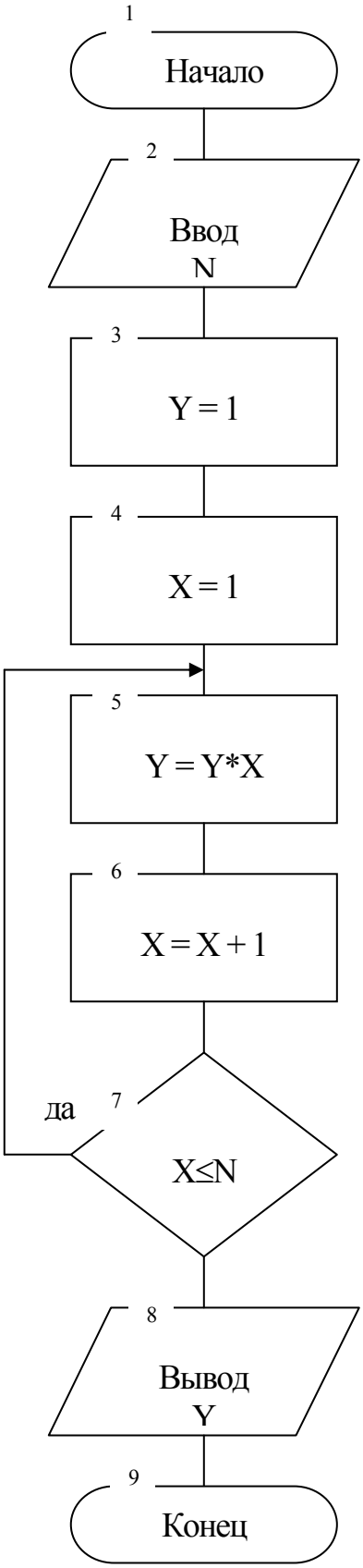
$$Y = N!$$

Два варианта схемы алгоритма вычисление факториала приведенные на рис. 5.

Итерационные циклы. При реализации итерационных вычислительных процессов в алгоритмах должно обеспечиваться обязательное выполнение условия выхода из цикла, т-є сходимость итерационного процесса.

Примером итерационных вычислительных процессов есть вычисление бесконечных числовых рядов. При этом, для практических расчетов ограничиваются вычислением некоторого числа элементов, исходя из требований заданной точности вычисления заданной суммы членов ряда S.

Числовой ряд, который сходится - это ряд, каждый следующий член которого имеет значение меньшее значение предыдущего члена ряда. В этом случае сумма членов ряда является конечной величиной. Вычисление суммы членов ряда прекращается на очередном члене ряда, значение которого меньше заданной точности.



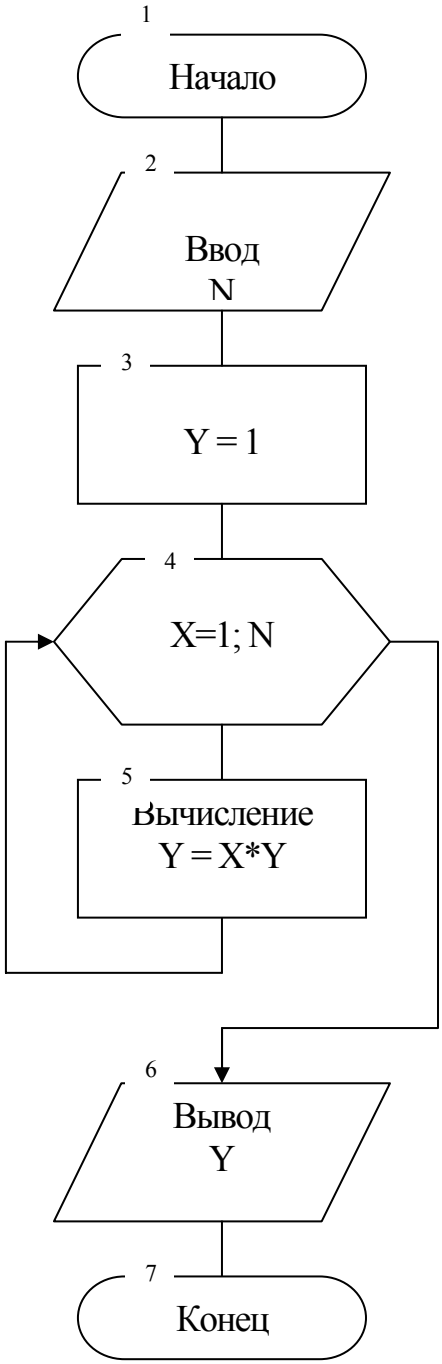


Рис. 5 Алгоритми вычисления факториала

Итерационные алгоритмы для вычисления сумм бесконечных рядов строятся в следующем порядке:

- вводятся необходимые исходные данные;

- задаются начальное значение суммы и вспомогательных переменных (если это необходимо);

- вычисляется значение текущего члена ряда;

- выполняется сравнение значения текущего члена ряда с заданной точностью ε ;

- если значение члена ряда не меньше заданной точности ε , то он добавляется к накопленной сумме и изменяются значения вспомогательных переменных, после чего осуществляется переход на вычисление очередного члена ряда и цикл повторяется;

- если значение текущего члена ряда меньше заданной точности ε , то осуществляется выход из цикла и выводится полученный результат.

В алгоритмах, которые реализуют итерационные вычислительные процессы, недопустимое использование блоков модификации, так как отсутствующая управляющая переменная - параметр цикла.

Пример: составить алгоритм для вычисления суммы сходящегося ряда с точностью ε

$$S = 1 + x + \frac{x^2}{2} + \frac{x^3}{3} + ... + \frac{x^n}{n} + ...$$

На схеме алгоритма (рис. 6) в блоке 3 задаются исходные значения номера **n** члена ряда, который вычисляется, и начальное значение суммы членов ряда **S**. В данном случае n = 1 и S = 1, то есть вычисление начинаются со второго члена ряда, так как первый член ряда, равный 1, невозможно вычислить по общей формулой члена ряда. Накопление суммы выполняется в блоке 5 с помощью рекурсивной зависимости

S = S + Y

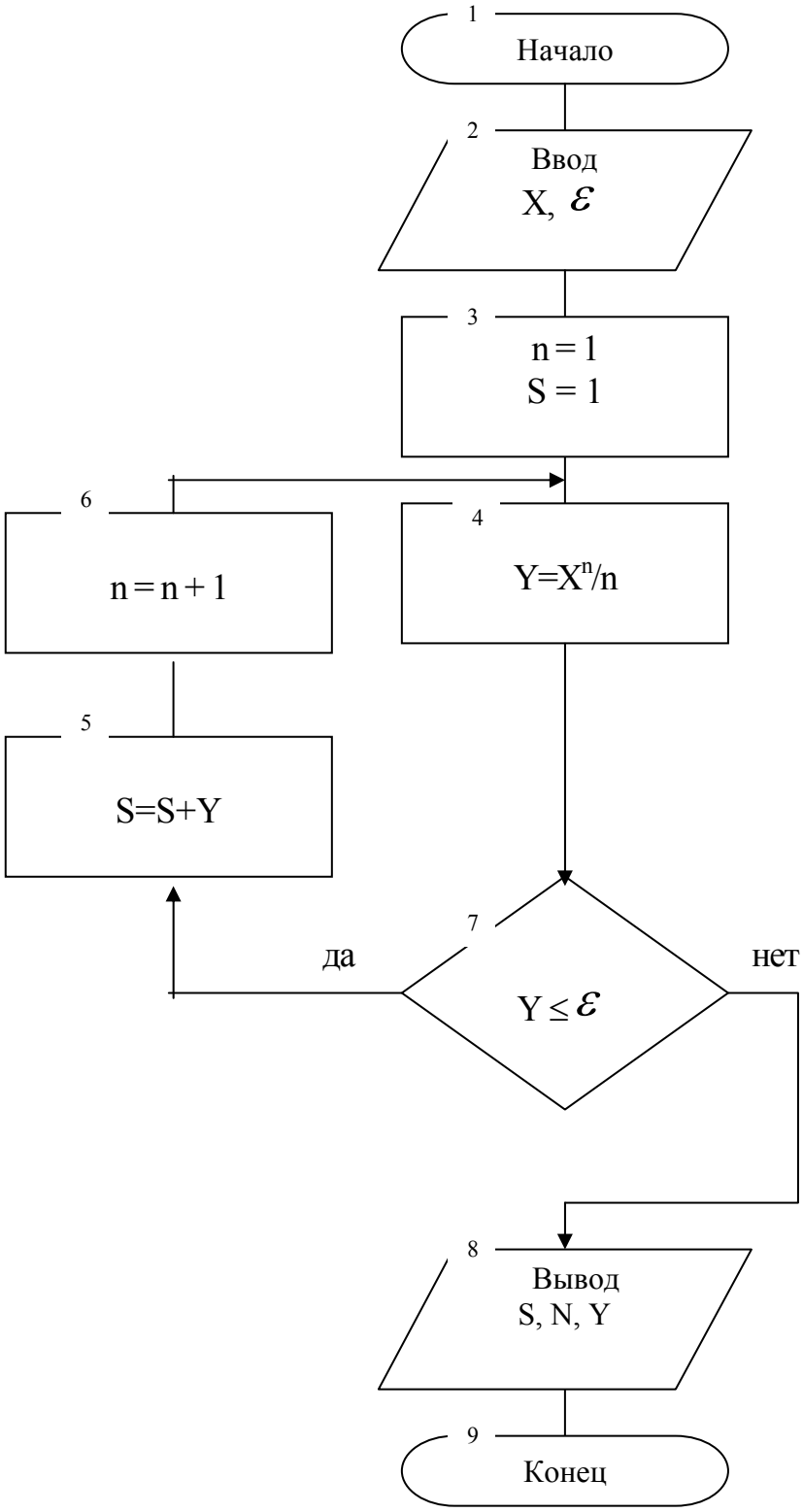


Рис. 6 Алгоритм вычисления суммы бесконечного ряда

Вложенные циклы. Внутри одного цикла могжет находиться однин или несколько других циклов. Цикл, содержащий в себе другой цикл, имеет название внешнего. Цикл, содержащийся в теле другого цикла, имеет название внутреннего. Основное правило построения вложенных циклов - этот охват внешним циклом внутреннего или нескольких внутренних. Глубина вложенности, то есть количество открытых циклов на участке алгоритма может быть больше двух и не ограничивается. Правила организации как внешнего, так и внутренних циклов аналогичны правилам организации простого цикла. Параметры циклов разных уровней изменяются не-одновременно.

Порядок изменения этих параметров определяется условиями задачи.

При организации внутренних циклов необходимо учесть, что область действия внутреннего цикла не должна выходить за область действия внешнего цикла.

Пример: составить алгоритм вычисления значения функции

$$Y = \sum_{i=1}^{N} \sum_{j=1}^{M} (i - j)^{i \bullet J}$$

В алгоритме (рис.7) несколько раз вычисляются суммы. Параметры циклов изменяются последовательно, то есть на одно значение параметра внешнего цикла параметр внутреннего цикла принимает последовательно все свои значения. Зафиксировав значения *i* во внешнем цикле, во внутреннем цикле производится накопление суммы при значениях *j,* меняющихся от 1 к M**.** После чего значение *i* увеличивается во внешнем цикле на 1 и внутренний цикл повторяется. При $i > N$, выполняется выход из цикла и выводится накопленный результат.

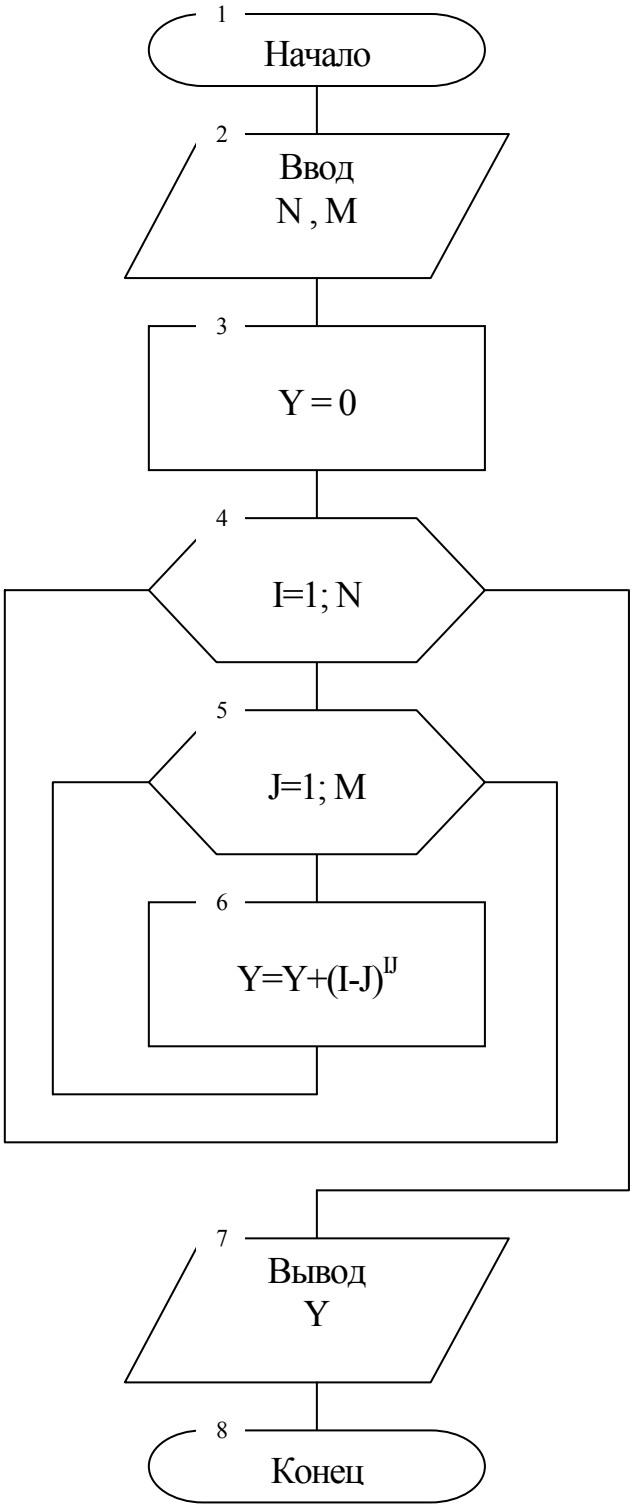


Рис. 7. Алгоритм вложенного циклического процесса