

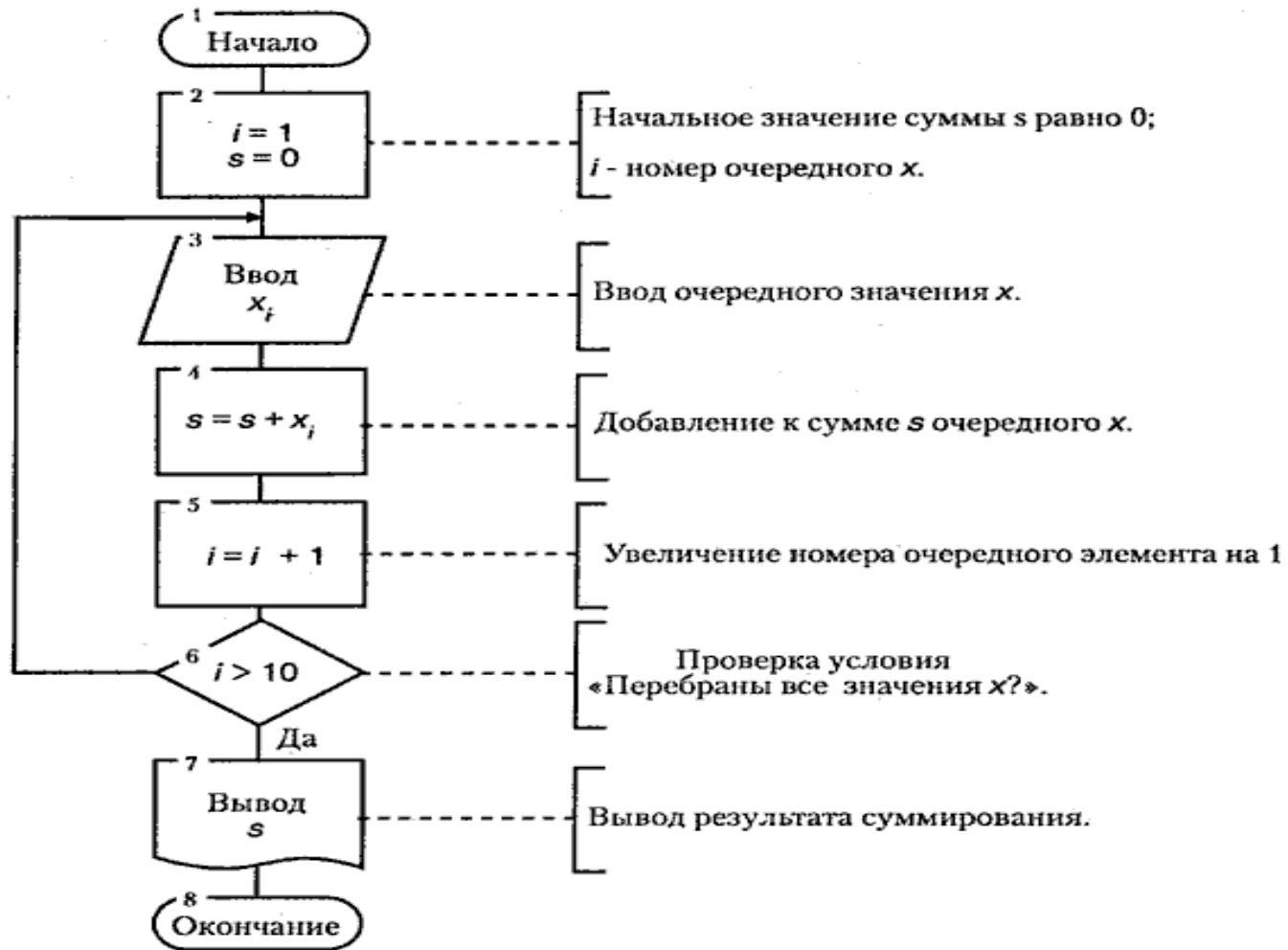
# Основы алгоритмизации и программирования

## Лекция 2

Блочное описание алгоритмов.

Виды алгоритмов.

# Пример блок - схемы





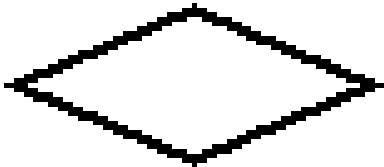


Алгоритм нахождения суммы 10-ти чисел

# Блоки на блок - схемах

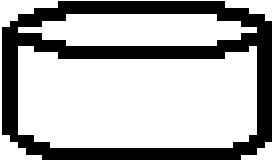




Операции обработки данных и носители информации изображаются на схеме соответствующими блоками.

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.

# Виды блоков

Наименование	Обозначение	Функции
Магнитный диск		Ввод-вывод данных, носителем которых служит магнитный диск.
Пуск-останов		Начало, конец, прерывание процесса обработки данных.
Соединитель		Указание связи между прерванными линиями, соединяющими блоки.
Межстраничный соединитель		Указание связи между прерванными линиями, соединяющими блоки, расположенные на разных листах.
Комментарий		Связь между элементом схемы и пояснением.

# 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.



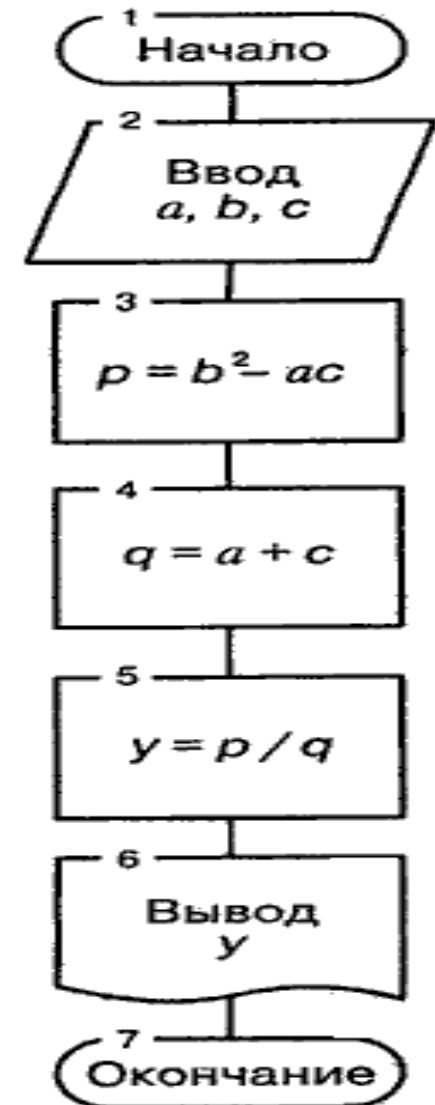
# Линейные алгоритмы

*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.

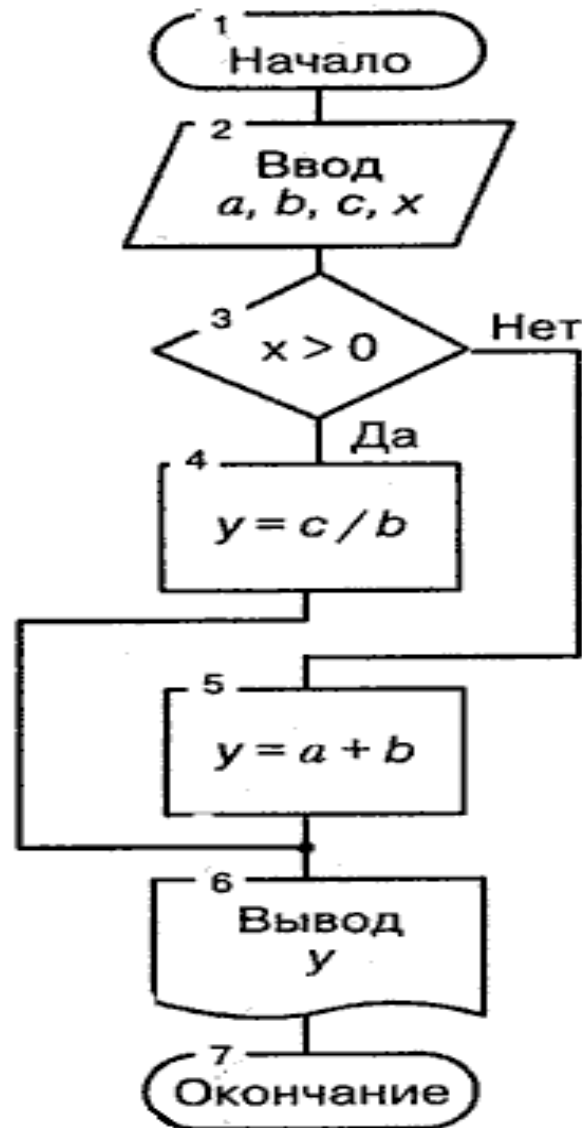
Следует иметь в виду, что, хотя на схеме алгоритма должны быть показаны все возможные направления вычислений в зависимости от выполнения определенного условия (или условий), при однократном прохождении программы процесс реализуется только по одной ветви, а остальные исключаются.

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:

$Y = (a+b)$ , если  $X < 0$ ;  
 $c/b$ , 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).

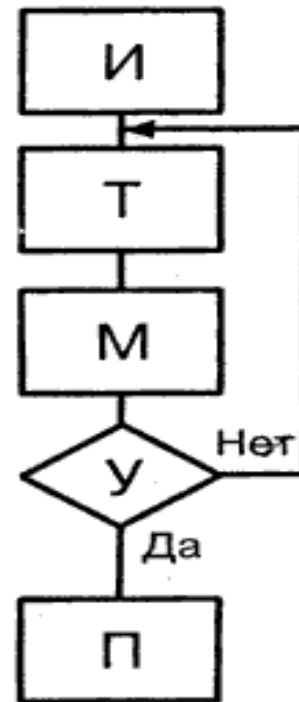
Порядок выполнения этих этапов, например, **Т** и **М**, может изменяться.

# Типы циклов

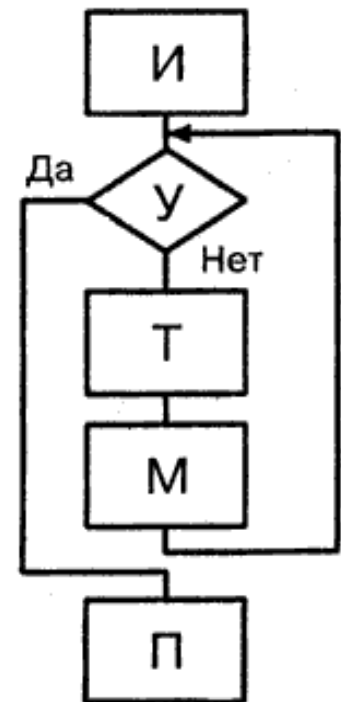
В зависимости от расположения проверки условия окончания цикла различают циклы с нижним и верхним окончаниями.

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

