

Алгоритмизация и программирование

Зайнуллин Айдар Фердинатович

Понятие алгоритма

Определение 1:

Алгоритм – это система точных и понятных предписаний о содержании и последовательности выполнения конечного числа действий, необходимых для решения любой задачи данного типа.

Определение 2:

Алгоритм – это правило, предписывающее последовательность действий над входной информацией и приводящее к результату.

Понятие алгоритма

Формальный характер предписаний (команд алгоритма), т.е. их независимость от содержания, вкладываемого в используемые в операциях числа, дает возможность их применения для любых исходных данных.

Ключевые понятия.

Команда – это указание исполнителю совершить некоторое действие.

Исполнитель (вычислитель) – устройство или живое существо, которое выполняет по определенным правилам составленный алгоритм. Исполнитель, который не понимает цели алгоритма, называется формальным исполнителем.

Свойства алгоритма

1. **Универсальность** (массовость) – применимость алгоритма к различным наборам исходных данных.
2. **Дискретность** – процесс решения задачи по алгоритму разбит на отдельные действия.
3. **Конечность** – каждое из действий и весь алгоритм в целом обязательно завершаются.
4. **Результативность** – по завершении выполнения алгоритма обязательно получается конечный результат.
5. **Выполнимость** (эффективность) – результата алгоритма достигается за конечное число шагов.
6. **Детерминированность** (определенность) – алгоритм не должен содержать предписаний, смысл которых может восприниматься неоднозначно. Т.е. одно и то же предписание после исполнения должно давать один и тот же результат.
7. **Последовательность** – порядок исполнения команд должен быть понятен исполнителю и не должен допускать неоднозначности.

Способы описания алгоритмов

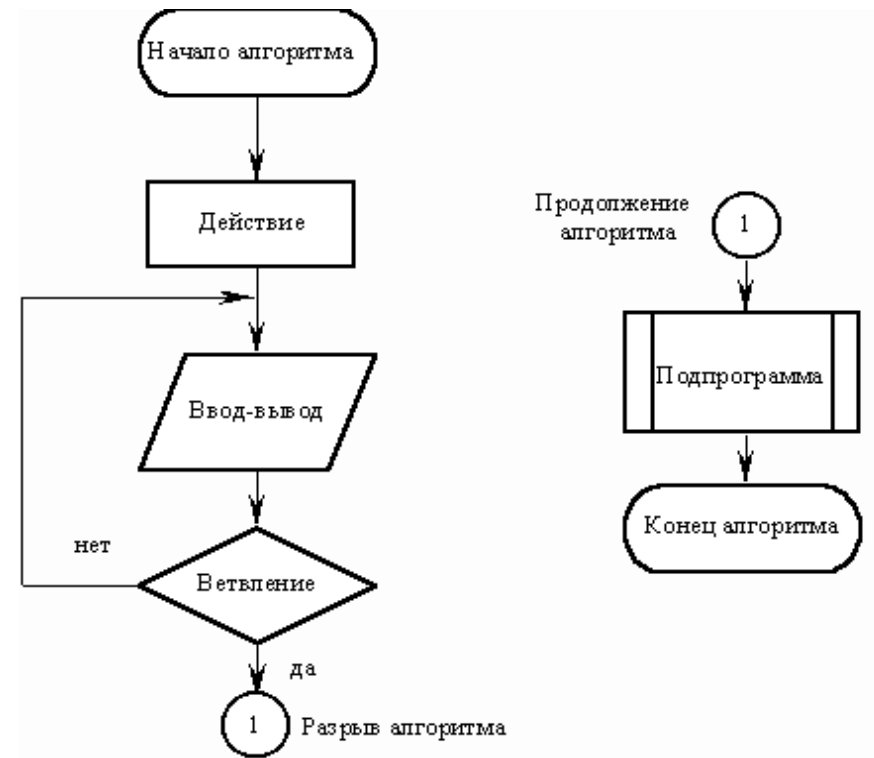
Вербальный (словесный), когда алгоритм описывается на человеческом языке

Графический, когда алгоритм описывается с помощью набора графических изображений

Символьный, когда алгоритм описывается с помощью специального набора символов (специального языка)

Словесная форма записи алгоритмов обычно используется для алгоритмов, ориентированных на исполнителя-человека. Команды такого алгоритма выполняются в естественной последовательности, если не оговорено противного.

Графическая запись с помощью блок-схем осуществляется рисованием последовательности геометрических фигур, каждая из которых подразумевает выполнение определенного действия алгоритма. Порядок выполнения действий указывается стрелками. Написание алгоритмов с помощью блок-схем регламентируется ГОСТом.

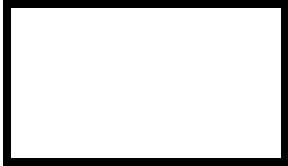

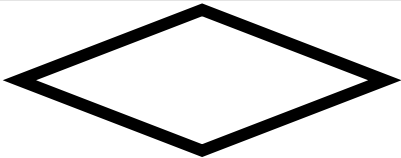




Блок-схемы

Преимущества:

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

Виды блоков

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

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

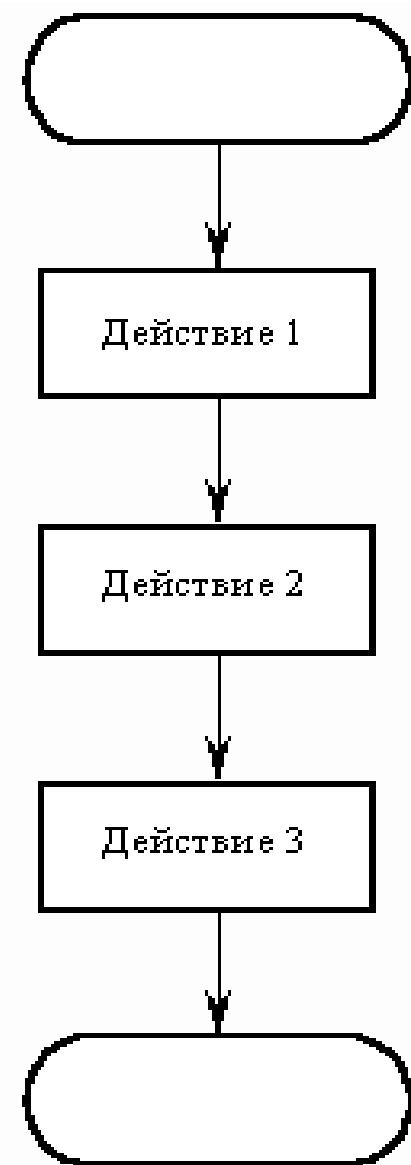
Любой вычислительный процесс может быть представлен как комбинация этих элементарных алгоритмических структур.

Виды алгоритмов

- линейные
- ветвящиеся
- циклические

Линейные алгоритмы

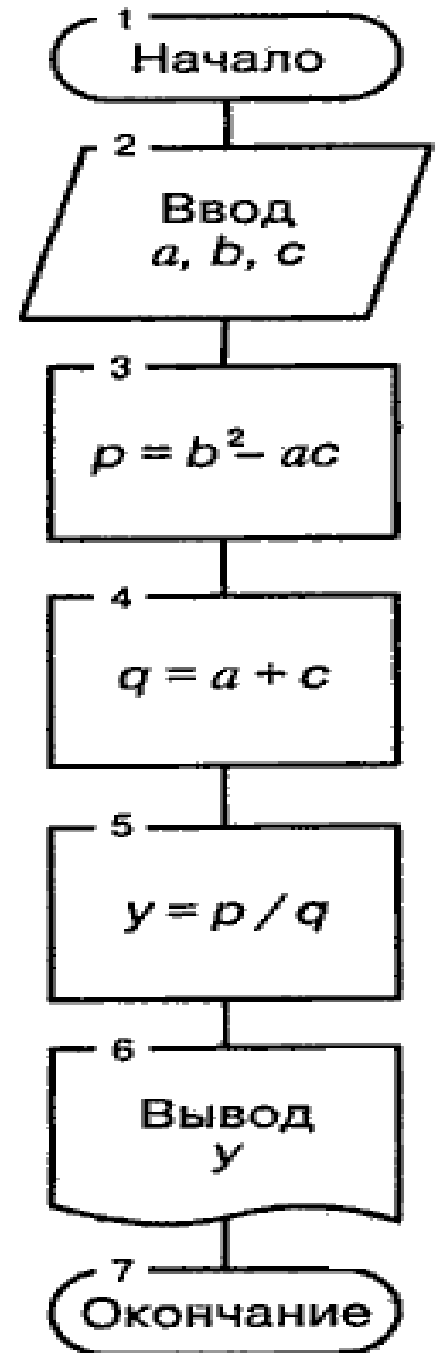
В линейном алгоритме операции выполняются последовательно, в порядке их записи. Каждая операция является самостоятельной, независимой от каких-либо условий. На схеме блоки, отображающие эти операции, располагаются в линейной последовательности.



Пример линейного алгоритма

Составить блок-схему алгоритма
вычисления арифметического
выражения

$$y = (b^2 - ac) / (a + c)$$



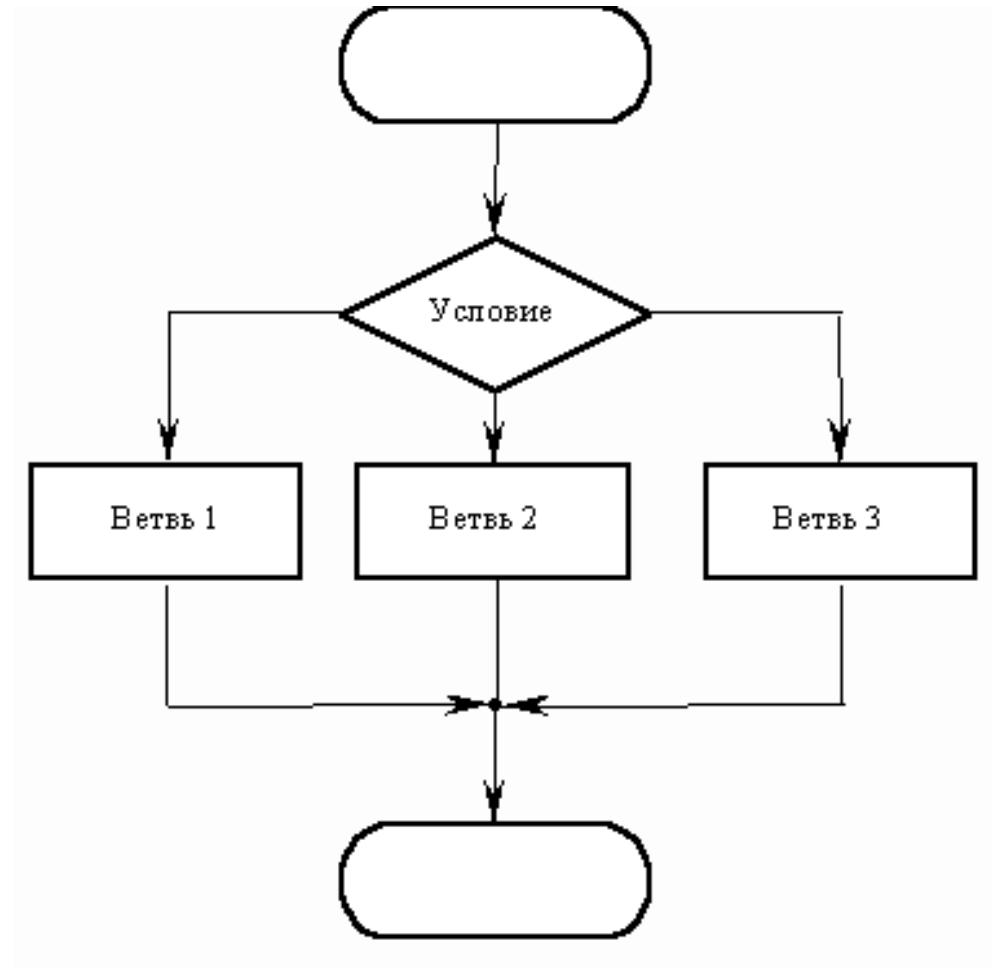
Алгоритм с ветвлением

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.

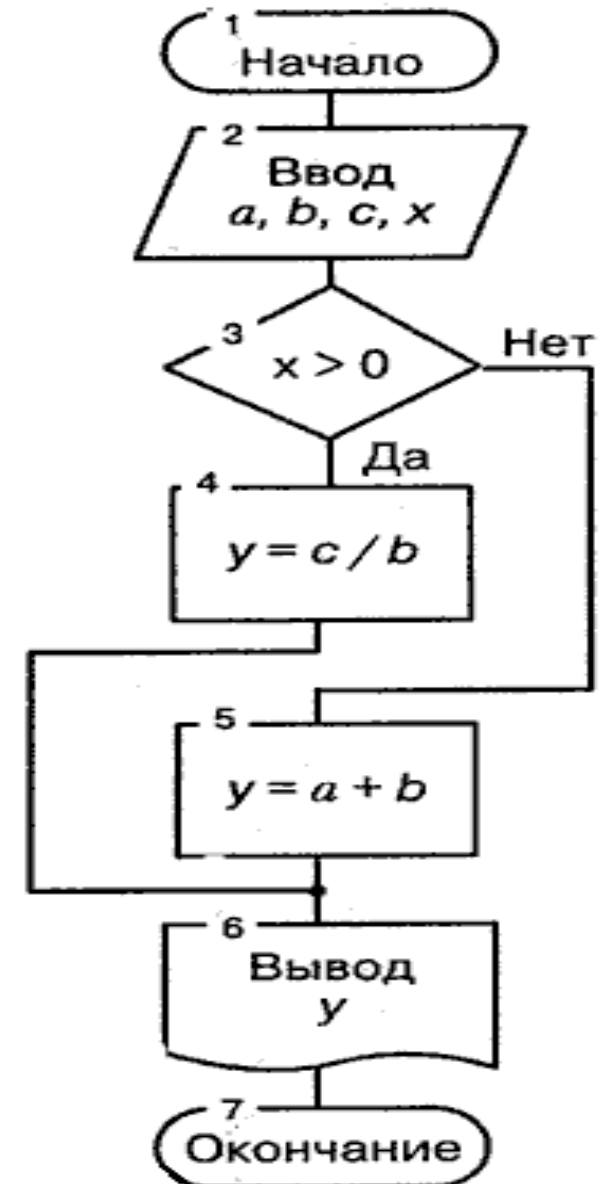
Следует иметь в виду, что, хотя на схеме алгоритма должны быть показаны все возможные направления вычислений в зависимости от выполнения определенного условия (или условий), при однократном прохождении программы процесс реализуется только по одной ветви, а остальные исключаются.

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:

$$Y = \begin{cases} (a+b), & \text{если } x < 0; \\ c/b, & \text{если } x > 0. \end{cases}$$



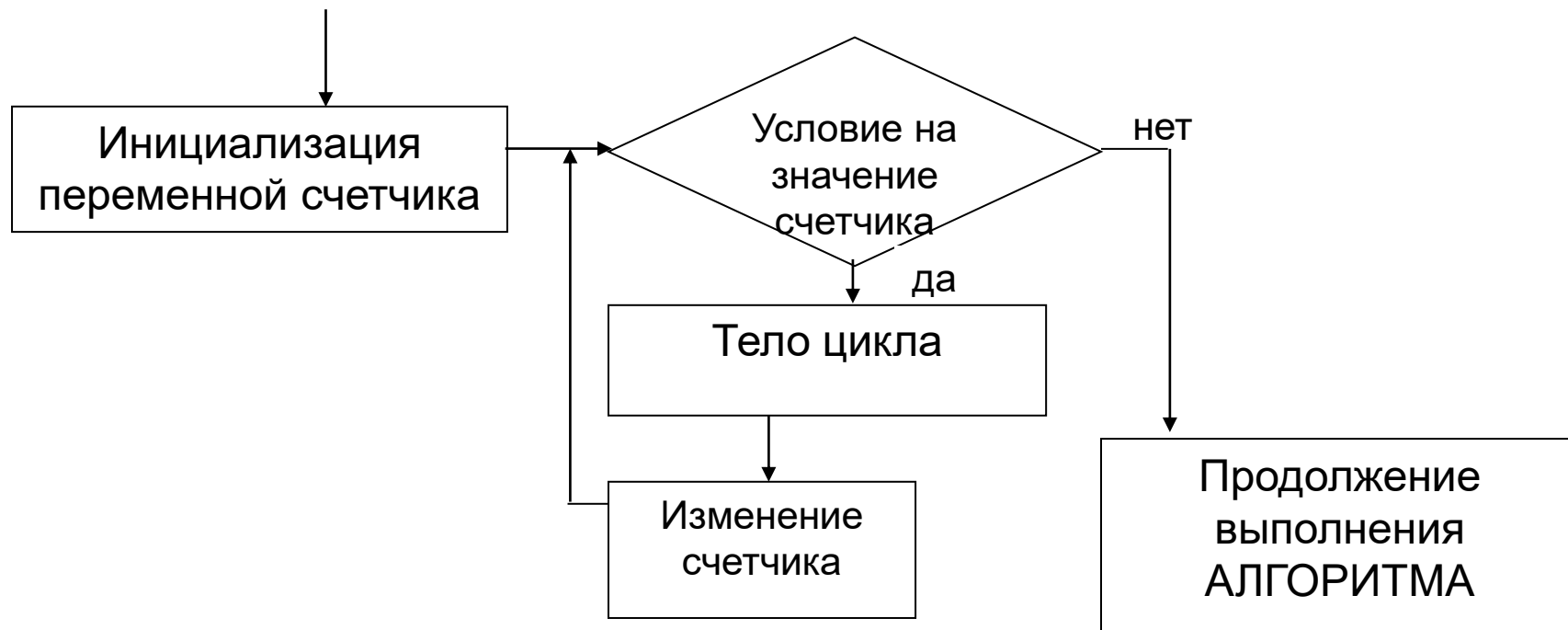
Циклические алгоритмы

Алгоритмы циклической структуры: в зависимости от выполнения или невыполнения какого-либо условия выполняется повторяющаяся последовательность действий, называемая телом цикла. Различают циклы с предусловием и постусловием:

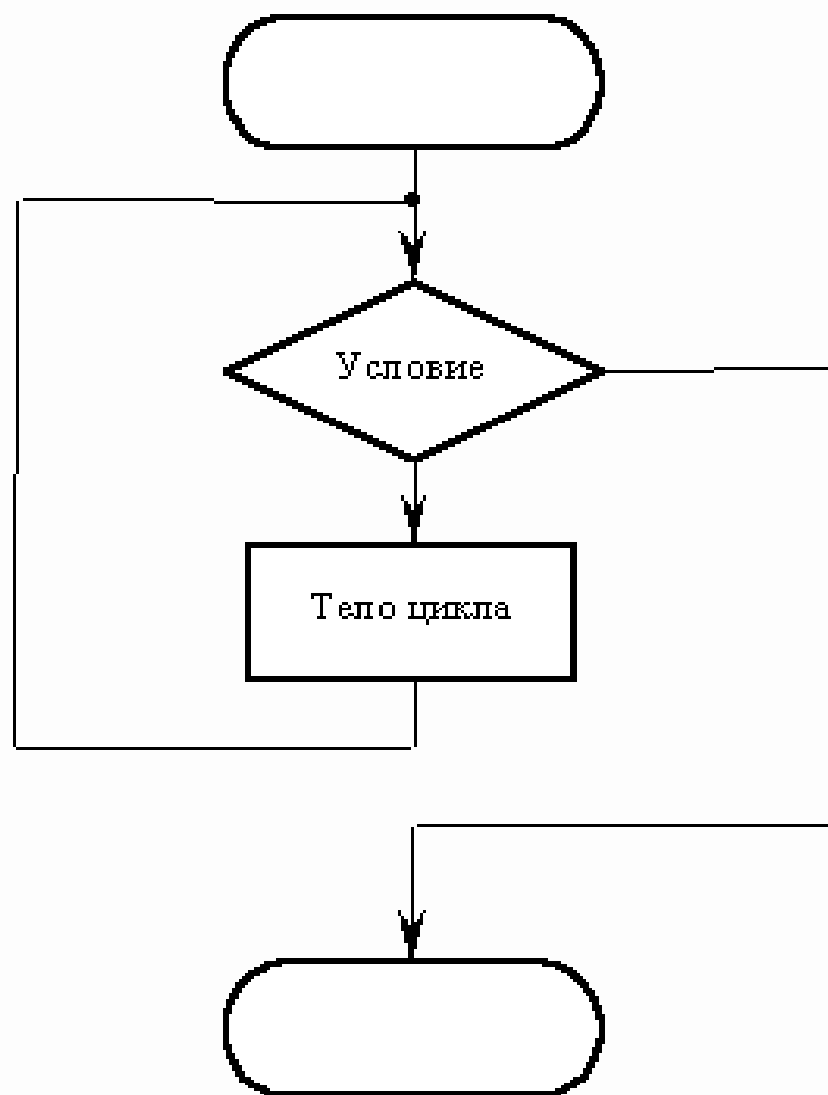
Этапы организации цикла

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

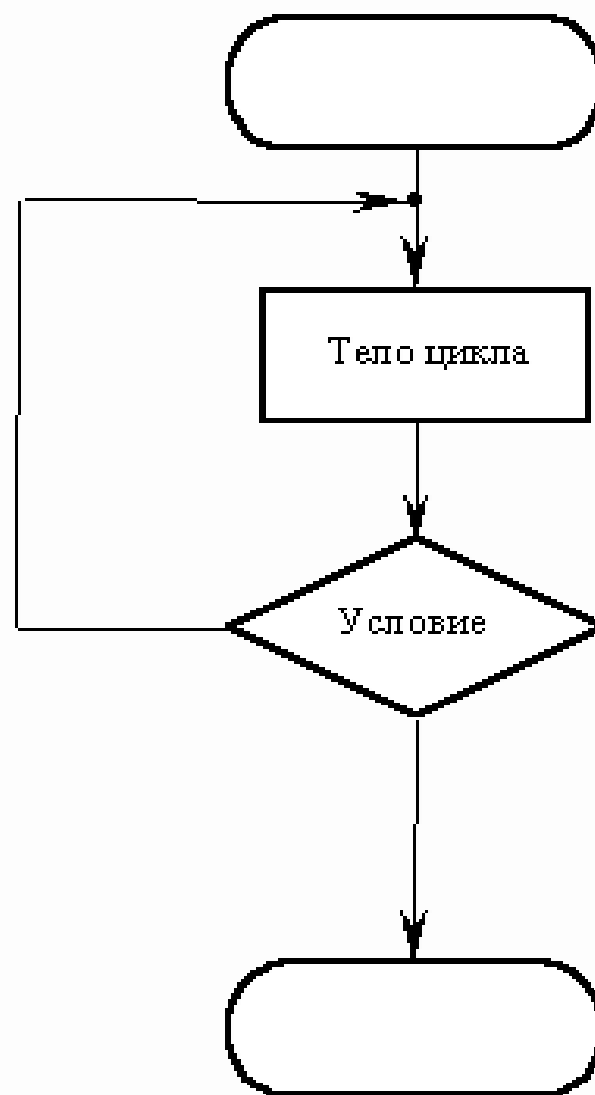
Языки программирования содержат операторы цикла со счетчиком. Они используются, когда изначально известно, сколько итераций (проходов) цикла необходимо выполнить. Модель цикла со счетчиком может быть описана с помощью классического цикла с предусловием.



Цикл с предусловием



Цикл с послеусловием



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

