

# **Процесс разработки программы**

# Разработку программы можно разбить на следующие этапы:

- 1) Составление алгоритма решения задачи.
- 2) Написание текста программы.
- 3) Отладка программы. Отладка программы — это процесс устранения ошибок из текста программы. Все ошибки делятся на синтаксические и логические. При наличии синтаксических ошибок (ошибок в написании операторов) программа не запускается. Логические ошибки — это ошибки, при которых программа работает, но неправильно.
- 4) Тестирование программы. Тестирование программы — процесс выявления ошибок в работе программы.

- *Алгоритм* - система четких однозначных указаний, которая определяет последовательность действий над некоторыми объектами и после конечного числа шагов приводит к получению требуемого результата.

# *Свойства алгоритмов*

- 1. Поочередное выполнение команд алгоритма за конечное число шагов приводит к решению задачи, к достижению цели. Разделение выполнения решения задачи на отдельные операции (выполняемые исполнителем по определенным командам) — важное свойство алгоритмов, называемое *дискретностью*.

# Свойства алгоритмов

- 2. Всякий алгоритм составляется в расчете на конкретного исполнителя с учетом его возможностей. Для того чтобы алгоритм мог быть выполнен, нельзя включать в него команды, которые исполнитель не в состоянии выполнить. Совокупность команд, которые могут быть выполнены исполнителем, называется *системой команд исполнителя*. Каждая команда алгоритма должна однозначно определять действие исполнителя. Такое свойство алгоритмов называется *определенностью (или точностью) алгоритма*.

# *Свойства алгоритмов*

- 3. Алгоритм, составленный для конкретного исполнителя, должен включать только те команды, которые входят в его систему команд. Это свойство алгоритма называется *понятностью*. Алгоритм не должен быть рассчитан на принятие каких-либо самостоятельных решений исполнителем, не предусмотренных составленным алгоритмом.

# *Свойства алгоритмов*

- 4. Еще одно важное требование, предъявляемое к алгоритмам, — *результативность (или конечность)* алгоритма. Оно означает, что исполнение алгоритма должно закончиться за конечное число шагов.

# *Свойства алгоритмов*

- 5. алгоритм должен быть *вариативен*, т. е. обеспечивать возможность решения для любых допустимых исходных значений. Такое свойство называется свойством массовости. Оно не является необходимым свойством алгоритма. Оно скорее определяет качество алгоритма

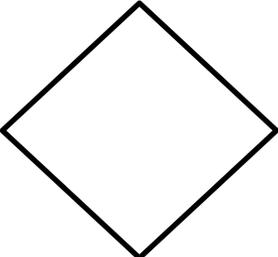
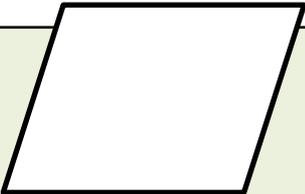
# Способы описания алгоритмов

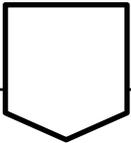
- Естественный язык
- Блок-схема
- Алгоритмический язык

- *Блок-схема* — графическое представление алгоритма. Каждый пункт алгоритма отображается на схеме некоторой геометрической фигурой — блоком — и дополняется элементами словесной записи

- Блоки на схемах соединяются линиями потоков информации. Основное направление потока информации идет сверху вниз и слева направо (стрелки могут не указываться), снизу вверх и справа налево — стрелка обязательна. Количество входящих линий для блока не ограничено. Выходящая линия должна быть одна (исключение составляет логический блок).

# **Основные элементы блок- схем**

Символ	Наименование	Описание
	Блок вычислений	Вычислительные действия или последовательность действий
	Логический блок	Выбор направления выполнения алгоритма в зависимости от некоторого условия
	Блоки ввода-вывода данных	Общие обозначения ввода (вывода) не зависимо от физического носителя
	Начало (конец)	Начала или конец алгоритма, вход или выход в программу

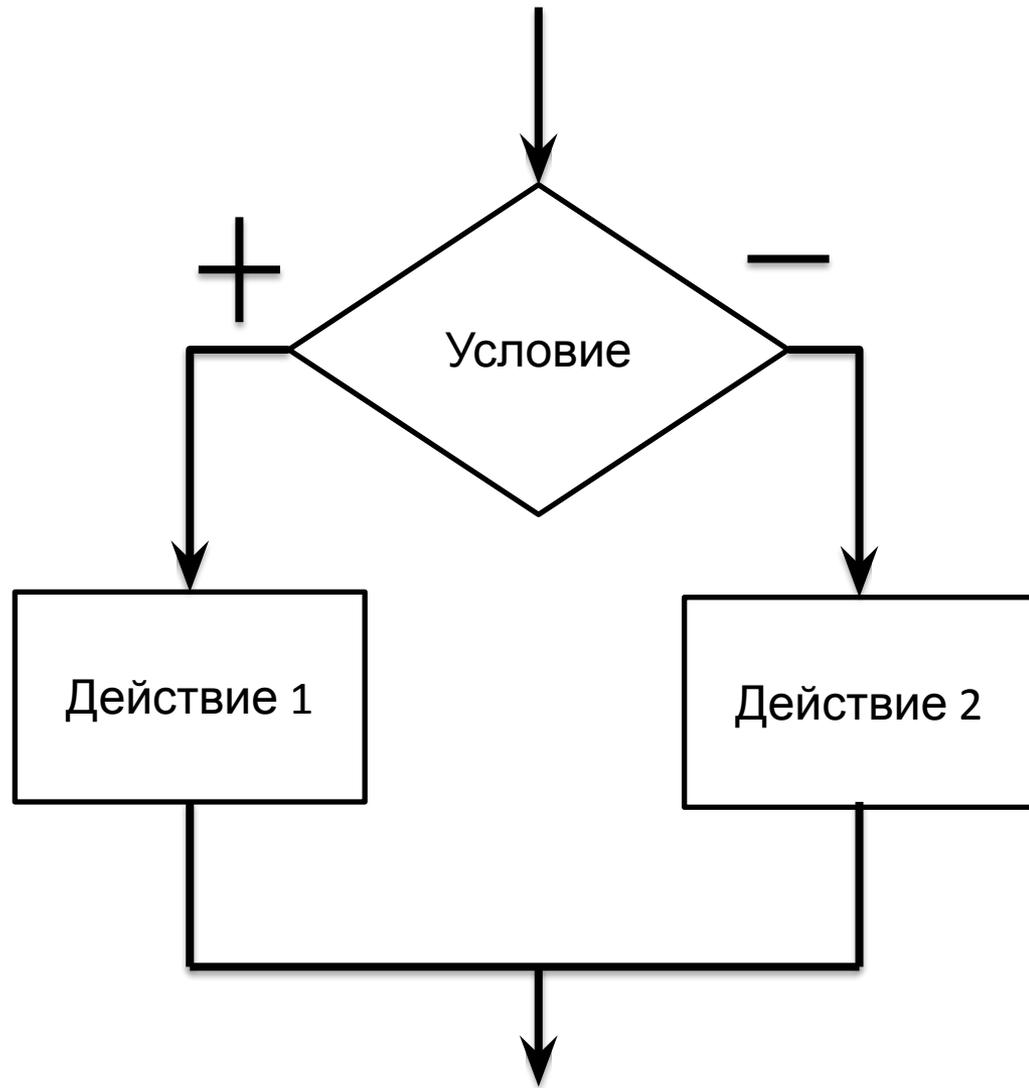
	<p><b>Процесс пользователя (подпрограмма)</b></p>	<p><b>Вычисление по стандартной программе или подпрограмме</b></p>
	<p><b>Блок модификации</b></p>	<p><b>Функция выполняет действия, изменяющие пункты алгоритма</b></p>
	<p><b>Соединитель</b></p>	<p><b>Указание связи между прерванными линиями потока информации в пределах одного листа</b></p>
	<p><b>Межстраничные соединения</b></p>	<p><b>Указание связи между информацией на разных листах</b></p>

# **Типовые алгоритмы программ**

- *Линейными* называются алгоритмы, в которых действия осуществляются последовательно друг за другом

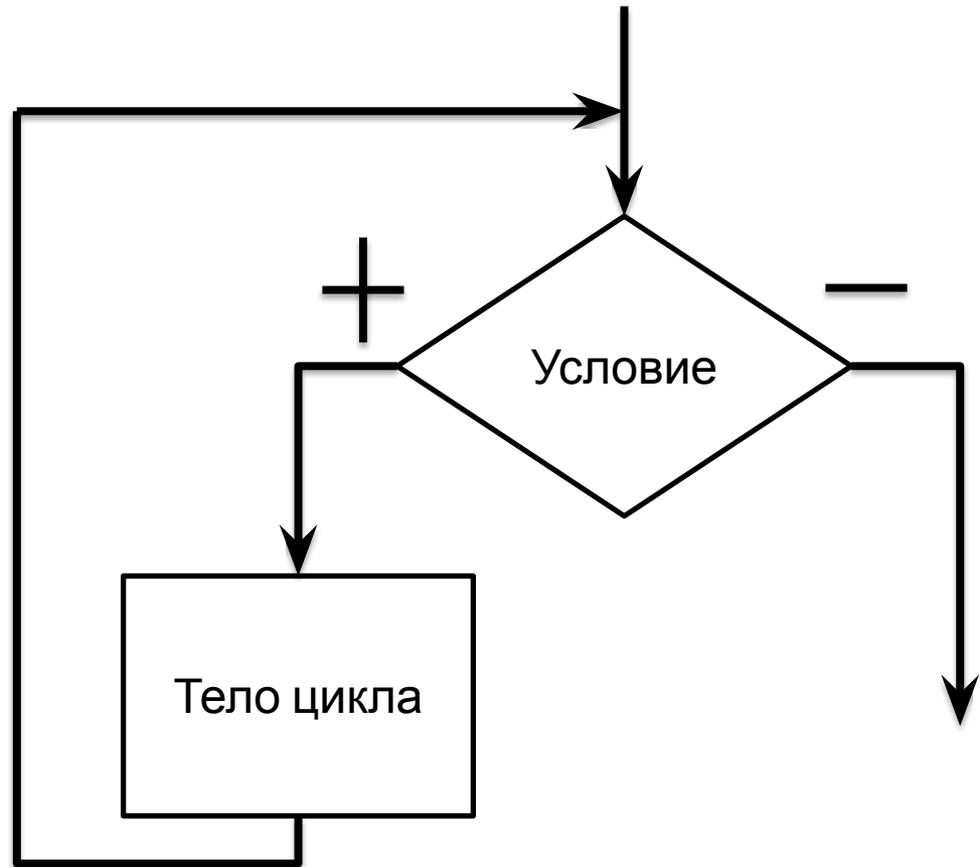


- *Разветвляющ*  
*имся*  
называется  
алгоритм, в  
котором  
действие  
выполняется  
по одной из  
ВОЗМОЖНЫХ  
ветвей  
решения  
задачи, в  
зависимости  
от выполнения  
условий.

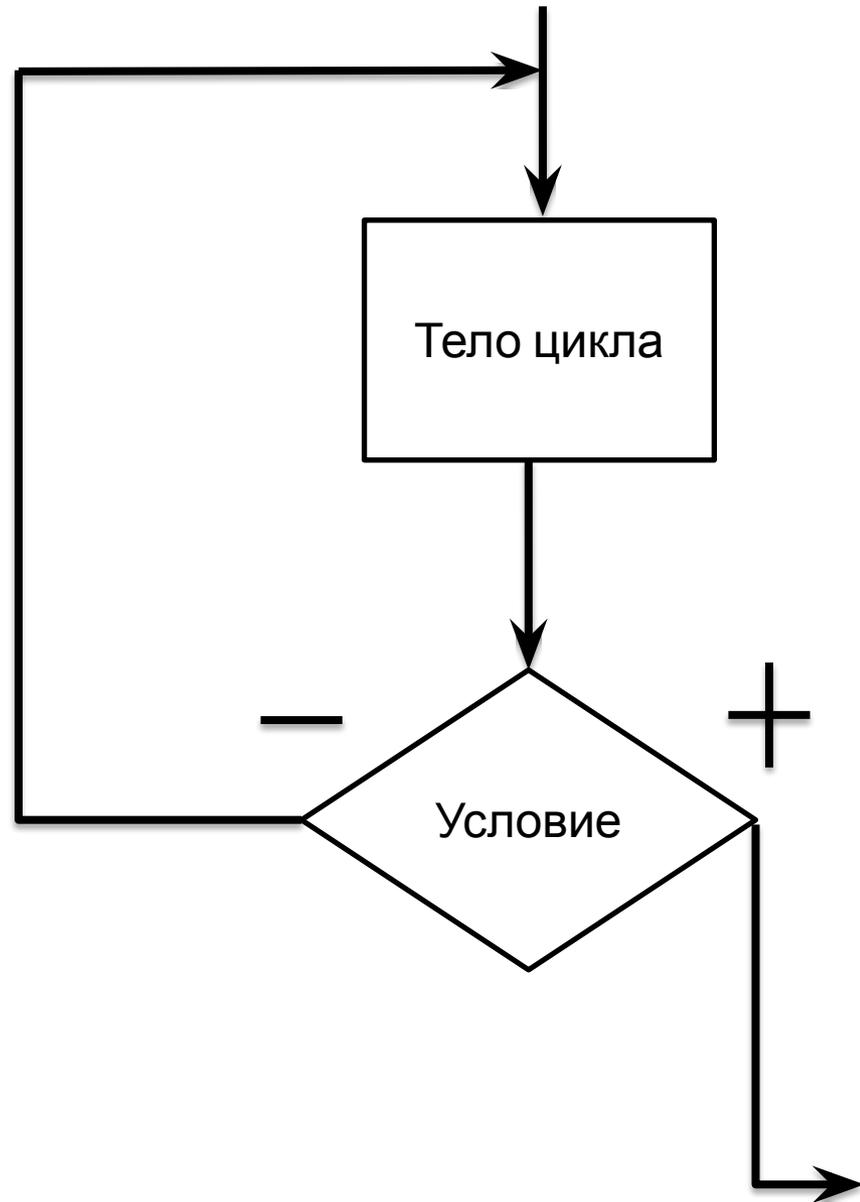


- *Циклическим* называется алгоритм, в котором некоторая часть операций (тело цикла) выполняется многократно. Циклические алгоритмы делятся на 3 типа:

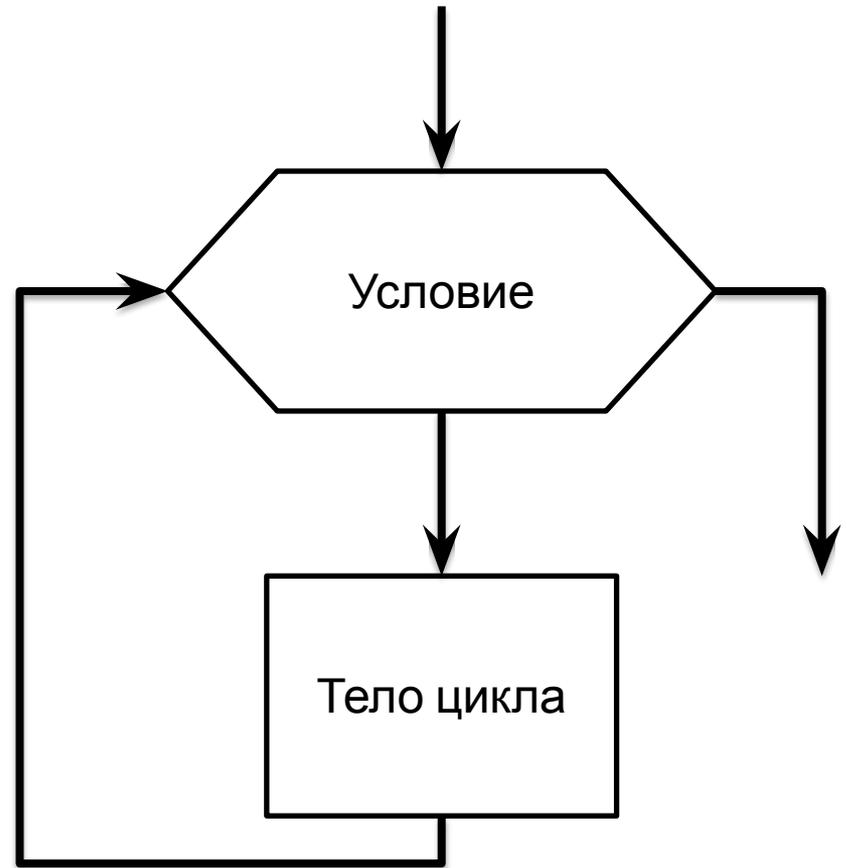
- Цикл с предусловием .  
Особенностью данного цикла является то, что тело цикла может не выполниться ни разу. Цикл прекращает выполнение как только условие становится ЛОЖНЫМ



- Цикл с постусловием . Его особенность – тело цикла выполнится минимум 1 раз. Цикл прекращает выполнение как только условие становится ИСТИННЫМ



- Цикл с параметром.  
Используется в том случае, когда известно точное количество шагов цикла.  
Является частным случаем цикла с предусловием



1. Составить блок-схему алгоритма сложения двух чисел (линейный)
2. Составить блок-схему алгоритма нахождения **минимального** из **двух** введенных чисел (разветвляющийся)
3. Составить блок-схему алгоритма нахождения **максимального** из **трех** введенных чисел (разветвляющийся)
4. Составить блок-схему алгоритма ввода с клавиатуры последовательности чисел, окончание ввода – ввод числа 0 (циклический)
5. Составить блок-схему алгоритма нахождения квадратов чисел от 1 до 5 (циклический)