

# *Алгоритмы.*



# Содержание.

- ❖ Что такое алгоритм?
- ❖ Свойства алгоритма.
- ❖ Форма записи алгоритмов.
- ❖ Базовые алгоритмические структуры.



# Что такое алгоритм?

Человек ежедневно встречается с необходимостью следовать тем или иным правилам, выполнять различные инструкции и указания. Например, переходя через дорогу на перекрестке без светофора надо сначала посмотреть направо. Если машин нет, то перейти полдороги, а если машины есть, ждать, пока они пройдут, затем перейти полдороги. После этого посмотреть налево и, если машин нет, то перейти дорогу до конца, а если машины есть, ждать, пока они пройдут, а затем перейти дорогу до конца.

В математике для решения типовых задач мы используем определенные правила, описывающие последовательности действий. Например, правила сложения дробных чисел, решения квадратных уравнений и т. д. Обычно любые инструкции и правила представляют собой последовательность действий, которые необходимо выполнить в определенном порядке. Для решения задачи надо знать, что дано, что следует получить и какие действия и в каком порядке следует для этого выполнить. Предписание, определяющее порядок выполнения действий над данными с целью получения искомых результатов, и есть алгоритм.

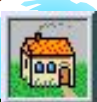
**Алгоритм** — заранее заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения решения задачи за конечное число шагов.

Название "алгоритм" произошло от латинской формы имени величайшего среднеазиатского математика Мухаммеда ибн Муса ал-Хорезми (Alhorithmi), жившего в 783—850 гг. В своей книге "Об индийском счете" он изложил правила записи натуральных чисел с помощью арабских цифр и правила действий над ними "столбиком", знакомые теперь каждому школьнику. В XII веке эта книга была переведена на латынь и получила широкое распространение в Европе. Понятие алгоритма является не только одним из главных понятий математики, но одним из главных понятий современной науки. Более того, с наступлением эры информатики алгоритмы становятся одним из важнейших факторов цивилизации

# Свойства алгоритма.

Основные свойства алгоритмов следующие:

- 1. Понятность** для исполнителя — исполнитель алгоритма должен понимать, как его выполнять. Иными словами, имея алгоритм и произвольный вариант исходных данных, исполнитель должен знать, как надо действовать для выполнения этого алгоритма.
- 2. Дискретность** (прерывность, раздельность) — алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).
- 3. Определенность** — каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких



**4. Результативность** (или конечность) состоит в том, что за конечное число шагов алгоритм либо должен приводить к решению задачи, либо после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.

**5. Массовость** означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

# Форма записи алгоритма.

На практике наиболее распространены следующие формы представления алгоритмов:

- словесная (запись на естественном языке);
- графическая (изображения из графических символов);
- псевдокоды (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- программная (тексты на языках программирования).

# Словесное описание алгоритма.

Словесный способ записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке

Словесный способ не имеет широкого распространения, так как такие описания:

- строго не формализуемы; страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний.



Например. Записать алгоритм нахождения **наибольшего общего делителя**

**(НОД)** двух натуральных чисел (алгоритм Эвклида).

Алгоритм может быть следующим:

1. задать два числа;
2. если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
3. определить большее из чисел;
4. заменить большее из чисел разностью большего и меньшего из чисел;
5. повторить алгоритм с шага 2.

Описанный алгоритм применим к любым натуральным



Назад

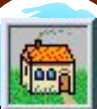
Далее

## Графическое описание алгоритма.

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным.

При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.

Такое графическое представление называется схемой алгоритма или блок-схемой. В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде блочного символа. Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий. В таблице приведены наиболее часто употребляемые символы.

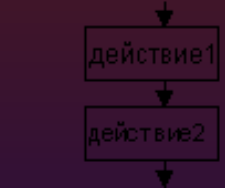
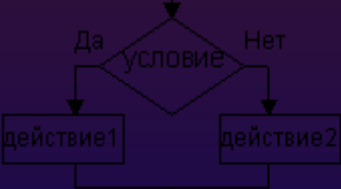
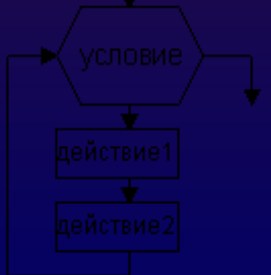


## Основные блоки для графической записи алгоритма.

вид блока	название / назначение	примеры записи
	<b>блок начала / конца алгоритма</b> обозначает начало или конец алгоритма	
	<b>блок ввода / вывода</b> служит для ввода исходных данных и вывода результатов	
	<b>блок действия</b> служит для записи команды присваивания	
	<b>блок логического условия</b> служит для организации ветвления в алгоритме	
	<b>блок цикла</b> служит для организации циклов в алгоритме	

# Базовые алгоритмические структуры.

Выделяют следующие виды алгоритмов: линейный, разветвляющийся, циклический и комбинированный. При определении вида алгоритма пользуются ключевыми словами.

Вид алгоритма.	Ключевые слова	Структура
<p>Алгоритм, в котором есть структура СЛЕДОВАНИЕ называется линейным. Следование – расположение действий друг за другом</p>	<p>Ключевых слов нет.</p>	<p><b>СЛЕДОВАНИЕ</b></p>  <pre> graph TD     Start(( )) --&gt; A[действие1]     A --&gt; B[действие2]     B --&gt; End(( ))             </pre>
<p>Алгоритм, в котором есть структура ВЕТВЛЕНИЕ называется разветвляющимся. Ветвление – выбор действия от какого-либо условия.</p>	<p>Если...то...иначе...; При...(в значении если)</p>	<p><b>ВЕТВЛЕНИЕ</b></p>  <pre> graph TD     Start(( )) --&gt; C{условие}     C -- Да --&gt; D[действие1]     C -- Нет --&gt; E[действие2]     D --&gt; Join(( ))     E --&gt; Join     Join --&gt; End(( ))             </pre>
<p>Алгоритм, в котором есть структура ЦИКЛ называется циклическим. Цикл – неоднократное повторение каких-либо действий.</p>	<p>От...до...; ...раз; Пока...; Если...(в значении пока...);</p>	<p><b>ЦИКЛ</b></p>  <pre> graph TD     Start(( )) --&gt; C{условие}     C --&gt; D[действие1]     D --&gt; E[действие2]     E --&gt; C     C --&gt; End(( ))             </pre>



Назад

Далее

# Примеры алгоритмов.

Линейный

Ветвление

Циклический

Назад

Далее

# Линейный алгоритм.



На рисунке представлен алгоритм лепки снеговика.

Назад



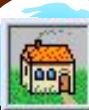
Далее

# Ветвление.



На рисунке продемонстрирован алгоритм ветвления на примере выбора пути маршрута.

Назад



Закончить

# Цикл.

На данном рисунке продемонстрирован циклический алгоритм на примере круговорота воды в природе.

