
АЛГОРИТМ

Лекция 6. Часть 2



Исполнитель алгоритма — это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.



Исполнителя характеризуют:

- среда;
- элементарные действия;
- система команд;
- отказы.



- Среда – это обстановка, в которой работает исполнитель.



СКИ

- Система команд исполнителя – набор понятных исполнителю команд.



Элементарное действие

После вызова команды исполнитель совершает элементарное действие

Отказы

Отказы возникают при вызове команды в недопустимом для данной команды состоянии среды.

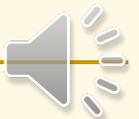


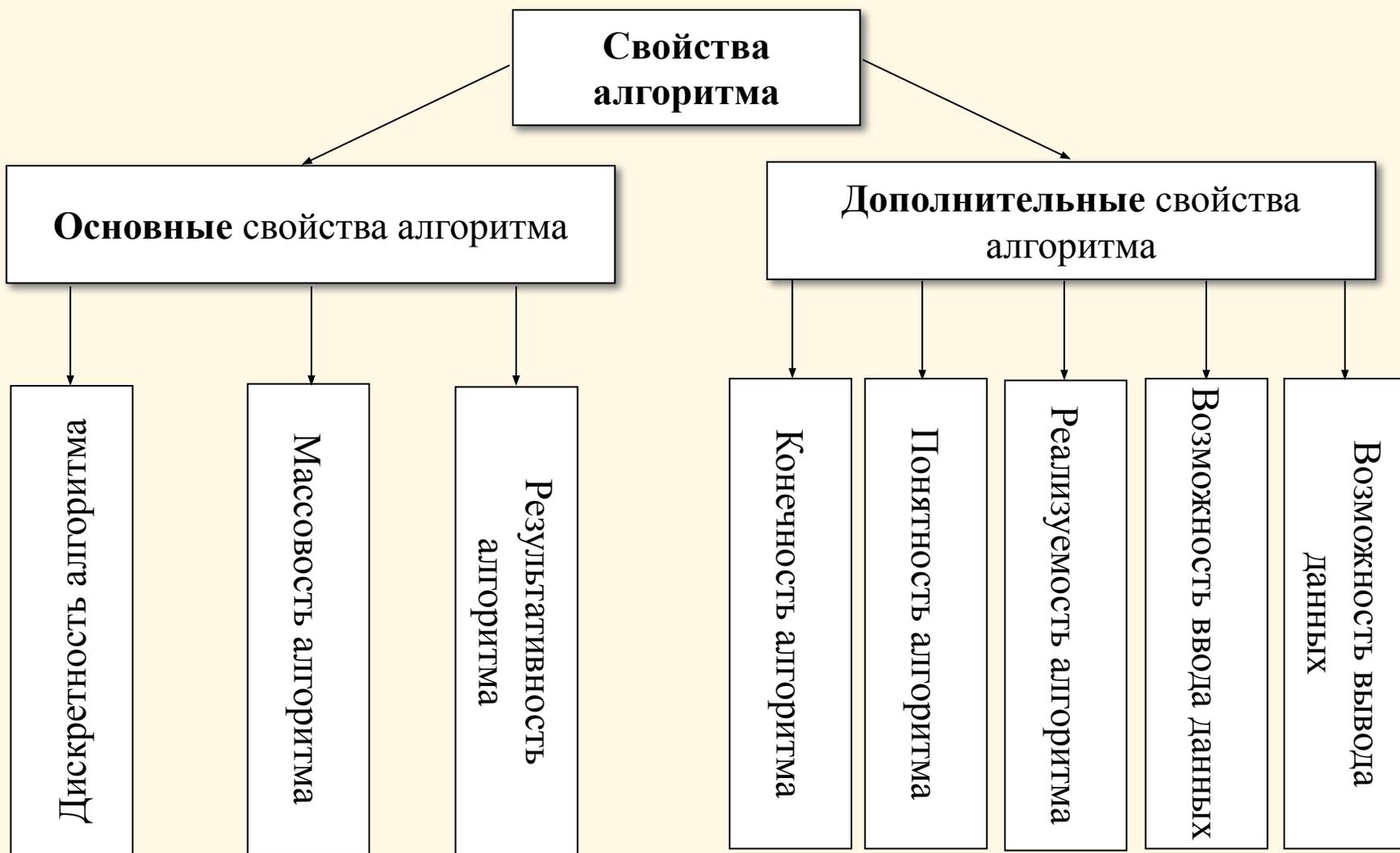
Основные свойства алгоритмов :

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.
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.





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).



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



Например, алгоритм Эвклида

Алгоритм нахождения **наибольшего общего делителя (НОД)** двух натуральных чисел

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.

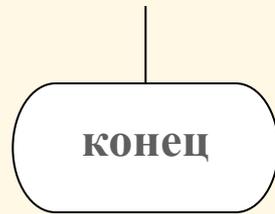


При **графическом** представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.

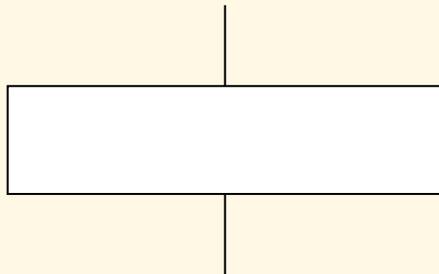


Блок-схемы

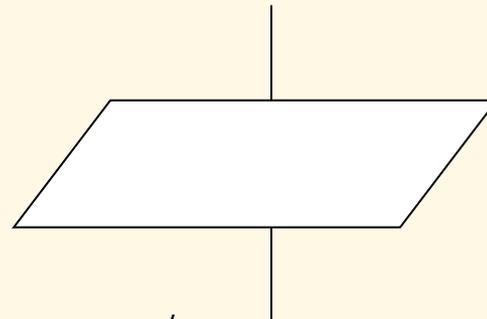
Основные геометрические фигуры:



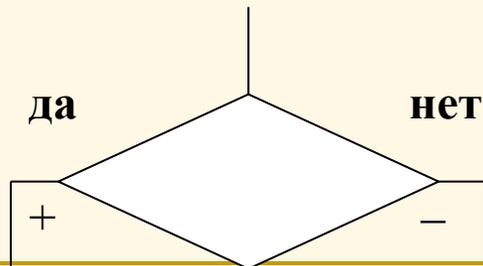
начало и конец алгоритма



блок действия

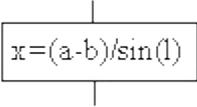
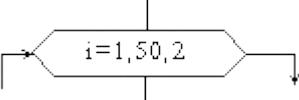
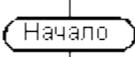


блок ввода/вывода данных

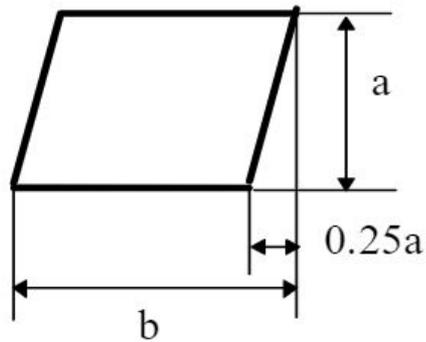


условный блок

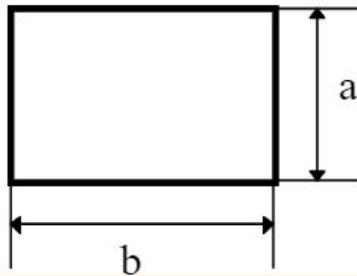


Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме, стандартной подпрограмме
Ввод-вывод		Ввод-вывод в общем виде
Пуск-останов		Начало, конец алгоритма, вход и выход в подпрограмму
Документ		Вывод результатов на печать

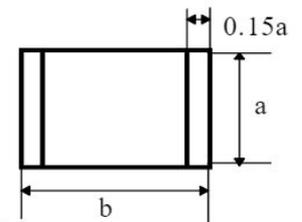
1. Данные
(ввод или вывод)



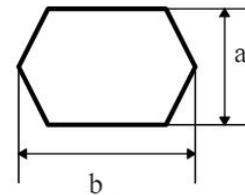
2. Процесс
(действие или
последовательность действий)



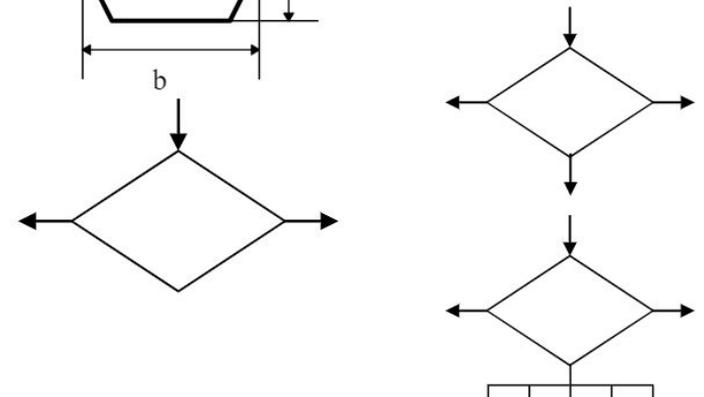
3. Предопределенный процесс
(вызов подпрограммы)



4. Подготовка



5. Решение



ГОСТ для описания схем алгоритмов

- ГОСТ 19.701-90 ЕСПД Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения.

Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов.

В псевдокоде не приняты строгие синтаксические правила для записи команд, присущие формальным языкам, что облегчает запись алгоритма на стадии его проектирования и дает возможность использовать более широкий набор команд, рассчитанный на абстрактного исполнителя.



Основные служебные слова

алг (алгоритм)	сим (символьный)	дано	для	да
арг (аргумент)	лит (литерный)	надо	от	нет
рез (результат)	лог (логический)	если	до	при
нач (начало)	таб (таблица)	то	знач	выбор
кон (конец)	нц (начало цикла)	иначе	и	ввод
цел (целый)	кц (конец цикла)	все	или	вывод
вещ (вещественный)	длин (длина)	пока	не	утв

Общий вид алгоритма: **алг** название
алгоритма (аргументы и результаты) **дано**
условия применимости алгоритма **надо**
цель выполнения алгоритма **нач** описание
промежуточных величин |
последовательность команд (тело
алгоритма) **кон**

