

Формы представления алгоритма

Алгоритмические конструкции

Алгоритм — описание последовательности действий (план), строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов.

Алгоритмизация — процесс разработки алгоритма (плана действий) для решения задачи.

Свойства алгоритмов

Дискретность (от лат. discretus — разделенный, прерывистый) — это разбиение алгоритма на ряд отдельных законченных действий (шагов).

Детерминированность (от лат. determinate — определенность, точность) - любое действие алгоритма должно быть строго и недвусмысленно определено в каждом случае.

Конечность - каждое действие в отдельности и алгоритм в целом должны иметь возможность завершения.

Массовость - один и тот же алгоритм можно использовать с разными исходными данными.

Результативность - в алгоритме не было ошибок.

Виды алгоритмов

Линейный (последовательный) алгоритм — описание действий, которые выполняются однократно в заданном порядке.

Циклический алгоритм — описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие. Перечень повторяющихся действий называется телом цикла.

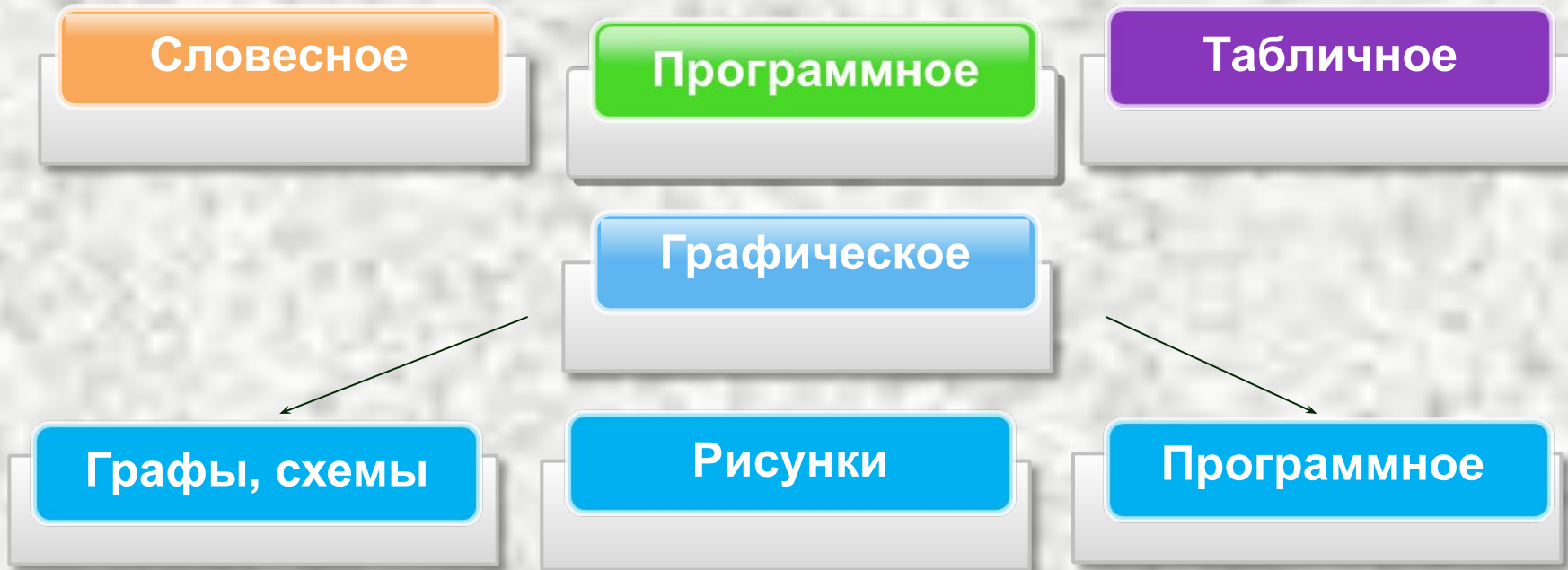
Разветвляющийся алгоритм — алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.

Вспомогательный алгоритм — алгоритм, который можно использовать в других алгоритмах, указав только его имя.

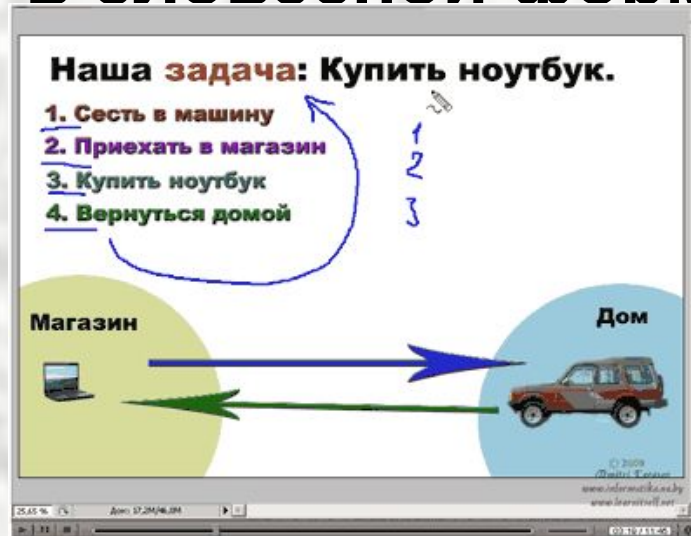
Формы представления алгоритмов

- Алгоритм может быть представлен в различных формах:
- словесной,
- графической,
- табличной,
- программной.

- **Способы представления алгоритмов**



Представление алгоритмов в виде описания последовательности действий, то есть в словесной форме



Такой способ представления несложен, но имеет недостатки. Главный недостаток состоит в том, что при таком способе допускается некоторая произвольность изложения, нет четких стандартов описания. Сложные задачи с анализом условий, с повторяющимися действиями и возвратами к предыдущим пунктам трудно представляются в словесном и словесно-формульном виде.

Графический способ представления алгоритмов



Одной из форм графического представления являются рисунки. Примеры представления алгоритмов в виде рисунков вы можете увидеть на упаковках продуктов быстрого приготовления, в инструкциях по использованию бытовой техники и пр.

Способ представления алгоритмов в виде графа



Граф — геометрический объект, состоящий из вершин и соединяющих вершины линий-дуг. В алгоритме анализа структуры предложения вершинами являются члены предложения, дуги показывают связи членов предложения, направления дуг — последовательность анализа (порядок действий алгоритма).


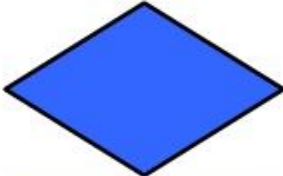




На рисунке представлен алгоритм «Разбор предложения» в виде графа.



Если алгоритм предназначен для исполнения техническим устройством, например станком с числовым программным управлением или компьютером, он представляется в виде программы.

Наиболее распространенной формой представления алгоритма является блок-схема.

Для отображения алгоритма в виде блок-схемы используется стандартный набор графических объектов (блоков), перечень и условные обозначения которых приведены в таблице.

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие (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

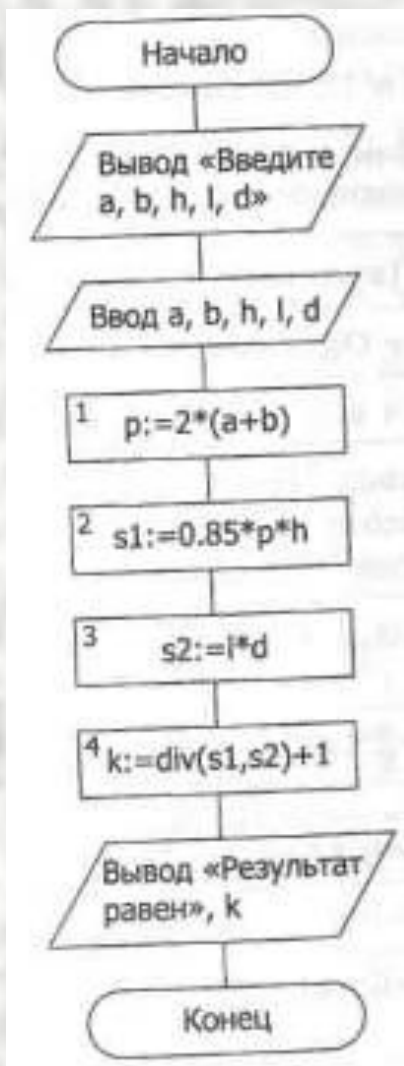
Приведем алгоритм решения задачи, представив его в разных формах.

Пример :

Требуется рассчитать необходимое количество рулонов обоев для оклейки комнаты. Заданы параметры комнаты: длина (a), ширина (b) и высота (h). Заданы параметры рулона обоев: длина (l), ширина (d). Считаем, что площадь окон и дверей составляет 15 % от площади стен.

- Словесно-формульное описание алгоритма «Оклейка обоями» представляется в виде нумерованной последовательности действий, понятных человеку.
- Алгоритм «Оклейка обоями»
- Рассчитать периметр комнаты: $p=2*(a+b)$.
- Рассчитать площадь стен с учетом дверей и окон: $s1=0,85*p*h$.
- Рассчитать площадь одного рулона обоев: $s2=l*d$.
- Вычислить количество рулонов: $k=\text{div}(s1/s2)+1$, где div — функция определения целой части числа.
- Конец алгоритма

Блок-схема алгоритма «Оклейка обоями»



Пояснения к блок-схеме:

- действия, указанные в блоках 1-4, соответствуют действиям, указанным в словесном алгоритме в пп. 1- 4;
- дополнительно введены блоки для ввода исходных данных в компьютер и вывода результата вычислений;
- дополнительно введены блоки начала и конца алгоритма.

Таблица Алгоритм «Оклейка обоями» в виде программы на школьном алгоритмическом языке

Школьный алгоритмический язык	Пояснения
алг Оклейка обоями	Начало алгоритма
нач вещ a, b, h, l, d, p, s1, s2, цел k	Описание типов переменных
вывод "Введите длину, ширину, высоту комнаты, длину, ширину обоев"	Вывод подсказки на экран
ввод a, b, h, l, d	Ввод информации с клавиатуры
$p := 2 * (a + b)$	Вычисление периметра комнаты
$s1 := 0.85 * p * h$	Вычисление площади стен
$s2 := l * d$	Вычисление площади рулона
$k := \text{div}(s1, s2) + 1$	Вычисление количества рулонов
вывод k КОН	Вывод ответа на экран Конец алгоритма

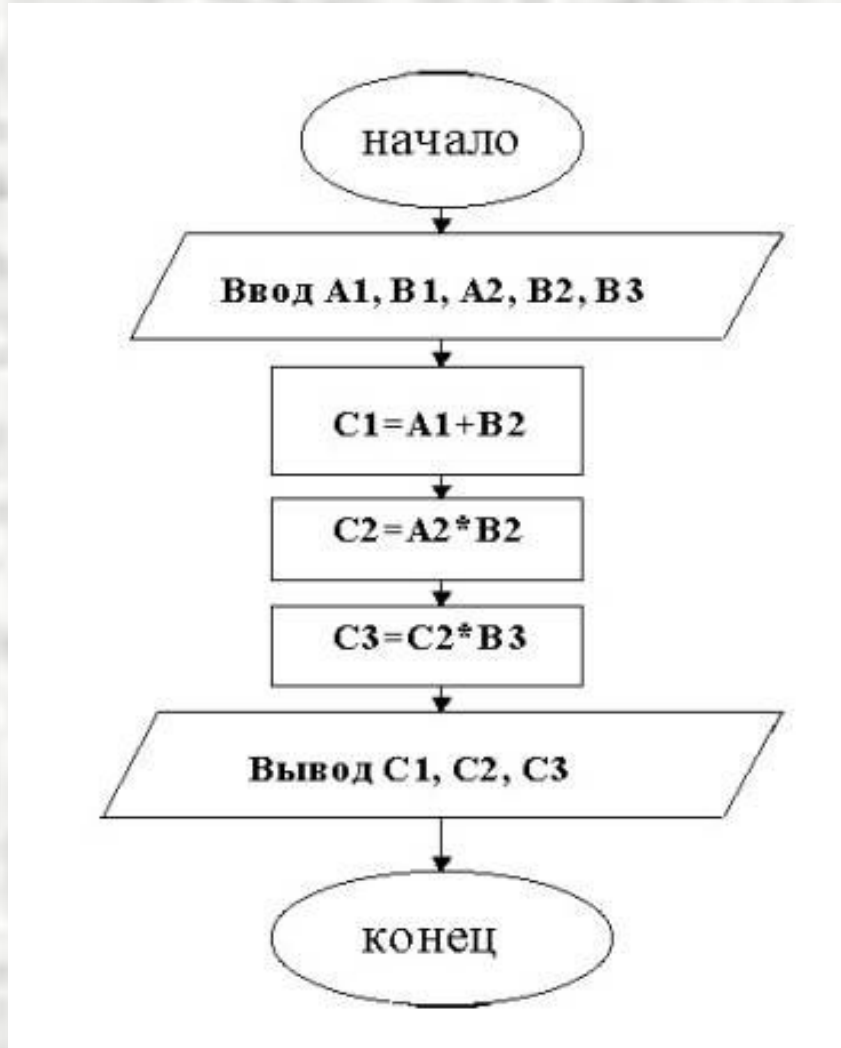
Алгоритмические конструкции

Любой, даже самый сложный алгоритм, можно представить с помощью трех типовых конструкций (структур):

- последовательности,
- ветвления,
- цикла.

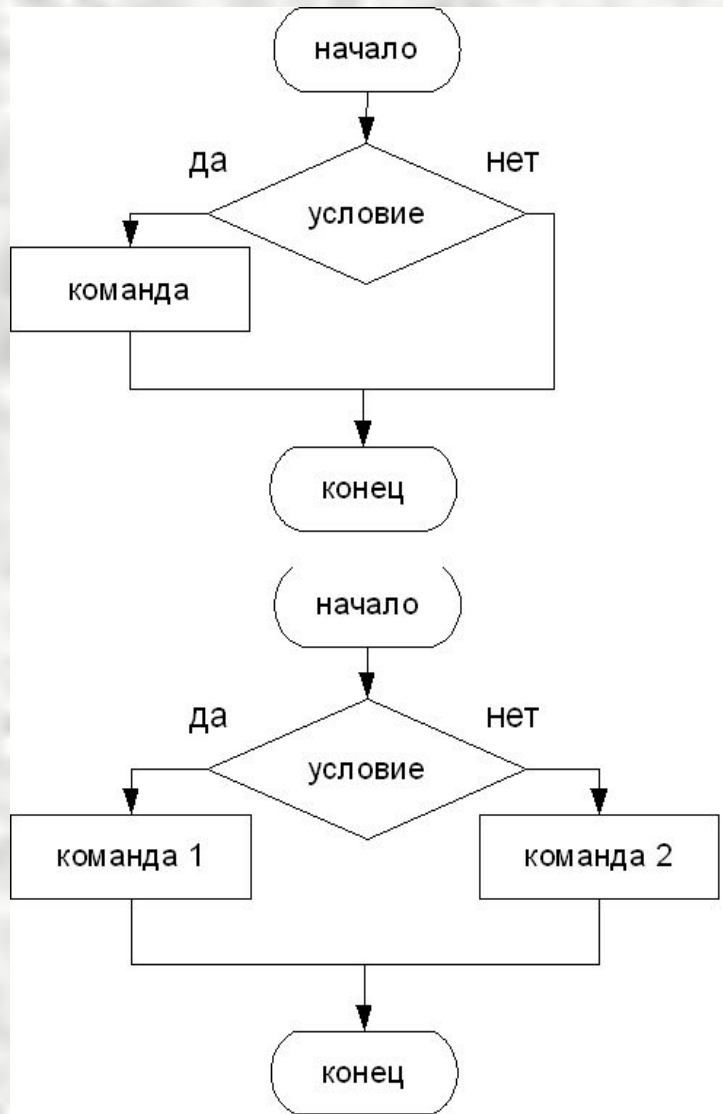
Каждая структура имеет один вход и один выход.

Блок-схемы базовых структур



В структуре «последовательность» действия выполняются последовательно, сверху вниз, без возвратов

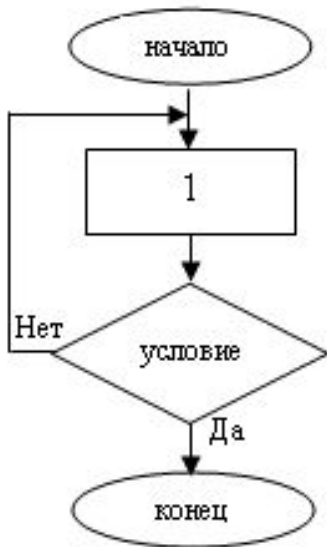
Блок-схемы базовых структур



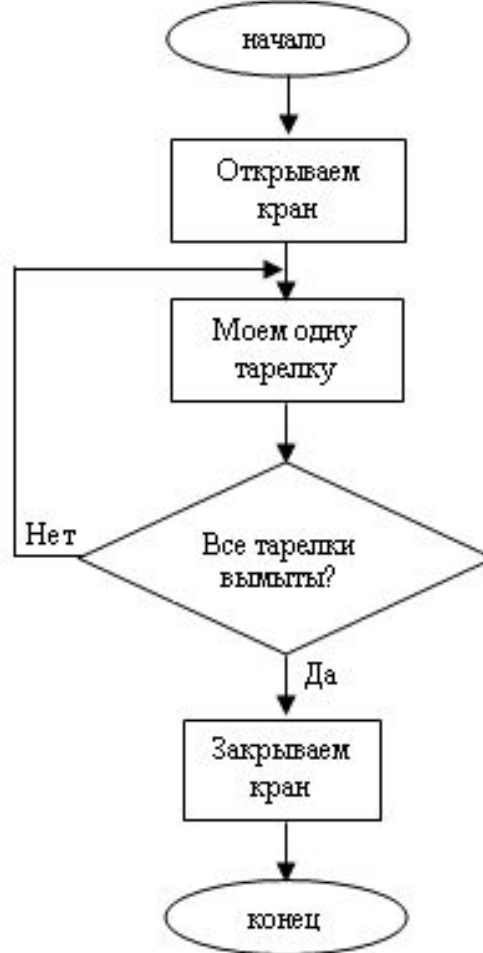
В структуре «ветвление» выполняется либо одна, либо другая группа действий в зависимости от истинности (выполнения) или ложности (невыполнения) условия

Блок-схемы базовых структур

Общий вид:



Пример: Кран



В структуре «цикл» действия повторяются до тех пор, пока выполняется заданное условие .

- Набор типовых структур часто называют алгоритмическими конструкциями, потому что из них, как из конструктора, можно составить алгоритм любой сложности.
- В зависимости от того, какие базовые структуры использованы при составлении алгоритмов, различают три основные разновидности алгоритмов: линейный, разветвляющийся, циклический и вспомогательный или подпрограмма.