

Понятие алгоритма,
способы описания,
свойства и типы
алгоритмов

Основные определения и понятия

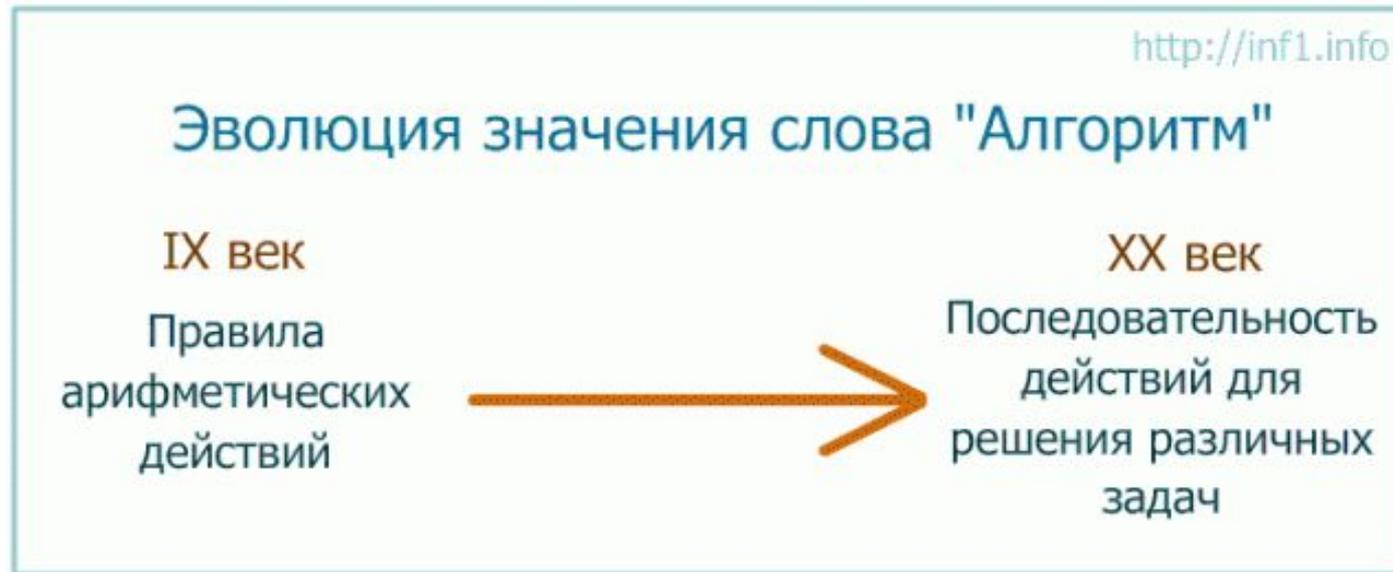
Алгоритмизация – это процесс построения алгоритма решения задачи, результатом которого является выделение этапов процесса обработки данных, формальная запись содержания этих этапов и определение порядка их выполнения.

Алгоритм – это точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату.

Полный перечень команд, который может выполнить исполнитель, называется **системой команд исполнителя**.

Исполнителем может быть как человек, так и любое техническое устройство.

Основные определения и понятия



Сначала определение понятия алгоритма было проблемой математики, однако с течением времени теория алгоритмов стала развиваться за счет влияния открытий не только в математике, но и в информатике. В настоящее время алгоритм является одним из главных понятий информатики.

Основные определения и понятия

Свойства алгоритма

- **дискретность**: состоит из отдельных шагов (команд); возможность расчленения вычислительного процесса на отдельные элементарные операции, возможность выполнения которых не вызывает сомнений
- **понятность**: должен включать только команды, известные исполнителю (входящие в СКИ)
- **детерминированность (определенность)**: при одинаковых исходных данных всегда выдает один и тот же результат; точность указаний, исключая их произвольное толкование
- **результативность (конечность)**: заканчивается за конечное число шагов

Основные определения и понятия

Свойства алгоритма

- **массовость**: может применяться многократно при различных исходных данных; пригодность алгоритма для решения всех задач заданного класса
- **корректность**: дает верное решение при любых допустимых исходных данных

Основные определения и понятия

Программа – это

- алгоритм, записанный на каком-либо языке программирования
- набор команд для компьютера

Команда – это описание действий, которые должен выполнить компьютер.

- откуда взять исходные данные?
- что нужно с ними сделать?

Основные определения и понятия

Программа – алгоритм, записанный в форме, воспринимаемой машиной.

Программа содержит наряду с описанием данных команды, в какой последовательности, над какими **данными** и какие операции должна выполнять машина, а также в какой форме следует получить результат.

Это обеспечивают различные **операторы**.

Средства изображения алгоритмов

Средства изображения алгоритмов

Основными изобразительными средствами алгоритмов являются следующие способы их записи:

- словесный;
- аналитический (формульно-словесный);
- блок-схемный;
- псевдокод;
- табличный;
- языки программирования.

Словесный способ

Словесный – содержание этапов вычислений задается на естественном языке в произвольной форме с требуемой детализацией.



- 1) отсутствует наглядность вычислительного процесса, т.к. нет достаточной формализации;
- 2) страдают многословностью записей;
- 3) допускают неоднозначность толкования отдельных предписаний.

Словесный способ

Пример

Алгоритм выполнения домашнего задания по математике.

- 1) открыть дневник;
- 2) посмотреть, что задано;
- 3) взять учебник и тетрадь по предмету;
- 4) прочитать параграф и выполнить письменные упражнения;
- 5) сравнить полученные результаты с ответами в конце учебника;
- 6) если ответы совпадают, закрыть тетрадь и учебник, если нет – повторить алгоритм с п. 4.

Словесный способ

Пример 2. Пусть задан массив чисел. Требуется проверить, все ли числа принадлежат заданному интервалу. Интервал задается границами А и В.

п.1 Берем первое число.

п.2 Сравниваем: выбранное число принадлежит интервалу; если да, то на **п.3**, если нет – на **п.6**.

п.3 Все элементы массива просмотрены? Если да, то на **п.5**, если нет – то на **п.4**.

п.4 Выбираем следующий элемент. На **п.2**.

п.5 Печать сообщения: все элементы принадлежат интервалу. На **п.7**.

п.6 Печать сообщения: не все элементы принадлежат интервалу. На **п.7**.

п.7 Конец.

Формульно-словесный способ

Формульно-словесный – задание инструкций с использованием математических символов и выражений в сочетании со словесными пояснениями.

Пример. Требуется написать алгоритм вычисления площади треугольника по трем сторонам.

п.1 – вычислить полупериметр треугольника $p=(a+b+c)/2$.

п.2 – вычислить $S = \sqrt{p(p-a)(p-b)(p-c)}$

п.3 – вывести S , как искомый результат и прекратить вычисления.

При использовании этого способа может быть достигнута любая степень детализации, более наглядно, но не строго формально.

Формульно-словесный способ

Алгоритм вычисления следующего выражения: $y = 2a - (x + 6)$.

- 1) ввести значения a и x ;
- 2) найти сумму $(x + 6)$;
- 3) найти произведение $(2 * a)$;
- 4) вычислить y как разность $y = 2a - (x + 6)$;
- 5) вывести y как результат вычисления выражения.

Псевдокод

Псевдокод - позволяет формально изображать логику программы, не заботясь при этом о синтаксических особенностях конкретного языка программирования. Обычно представляет собой смесь операторов языка программирования и естественного языка. Является средством представления логики программы, которое можно применять вместо блок-схемы.

Пример псевдокода

Алгоритм сложения двух чисел.

Псевдокод:

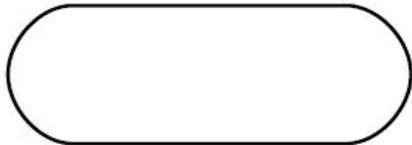
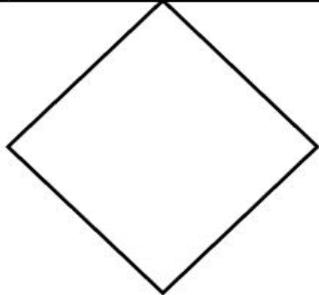
- 1) Ввод двух чисел a и b .
- 2) Вычисление суммы $S = a + b$.
- 3) Вывод S .
- 4) Конец.

Блок-схемный способ

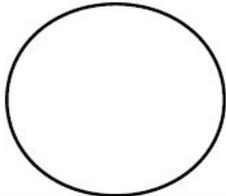
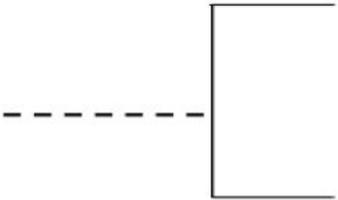
Блок-схемный – это графическое изображение логической структуры алгоритма, в котором каждый этап процесса переработки данных представляется в виде геометрических фигур (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций.

Блочные символы соединяются *линиями переходов*, определяющими очередность выполнения действий.

Основные символы блок-схем

Наименование	Символ	Выполняемые действия
Терминатор (пуск-остановка)		Обозначает начало и конец алгоритма
Данные (ввод-вывод)		Ввод и вывод данных
Процесс		Вычислительные действия
Решение		Наличие условия в алгоритме

ОСНОВНЫЕ СИМВОЛЫ БЛОК-СХЕМ

Наименование	Символ	Выполняемые действия
Граница цикла		Наличие цикла в алгоритме
Предопределенный процесс		Наличие подпрограммы
Соединитель		Разрыв алгоритма (соединительные блоки)
Комментарий		Внесение комментариев

Базовые канонические структуры алгоритмов

Доказано, что любую программу можно написать, используя комбинации трех управляющих структур:

- следование (линейный алгоритм);
- ветвление (разветвляющийся алгоритм);
- цикл (циклический алгоритм).

Программа, составленная из канонических структур, будет называться **регулярной программой**, т.е. иметь 1 вход и 1 выход, каждый оператор в программе может быть достигнут при входе через ее начало (нет недостижимых операторов и бесконечных циклов). Управление в такой программе передается сверху-вниз.

Снабженные комментариями, такие программы хорошо читабельны.

Линейные алгоритмы

Линейными называют алгоритмы, в которых операции выполняются последовательно одна за другой, в естественном и единственном порядке следования. В таких алгоритмах все блоки имеют последовательное соединение логической связью передачи информационных потоков. В алгоритмах с линейной структурой каждый оператор выполняется только один раз и выполняется весь целиком, т.е. выполняются все ее операторы.

В них могут использоваться все блоки, за исключением блоков проверки условия и модификации.

Линейные алгоритмы, как правило, являются составной частью любого алгоритмического процесса.

Линейные алгоритмы

Алг Дежурик

нач

1. Открыть окно
2. Вымыть тряпку
3. Вытереть доску
4. Принести мел
5. Заккрыть окно
6. Сесть на место

КОН



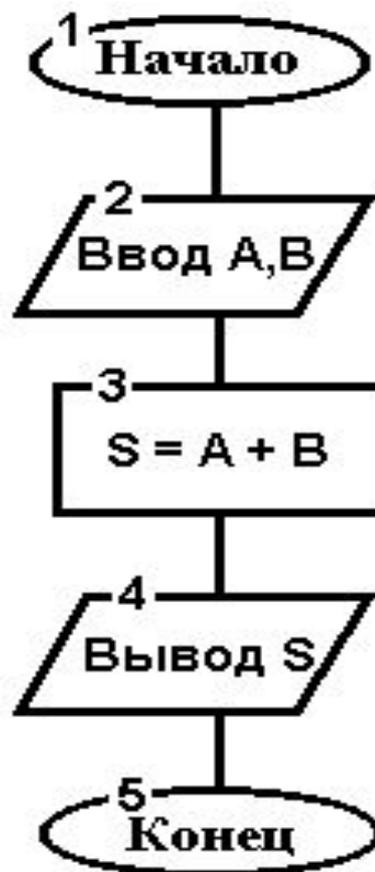
Линейные алгоритмы

Задача:

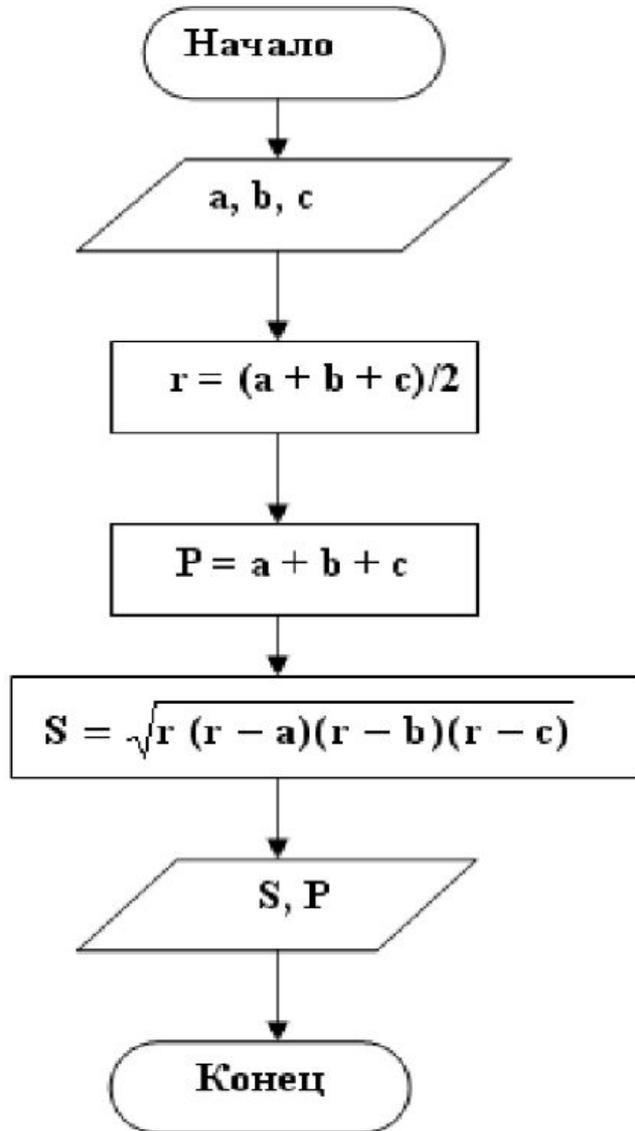
Даны A, B

Найти $S=A+B$

Схема алгоритма:



Линейные алгоритмы



Псевдокод

1. Ввод чисел a, b, c
2. Вычисление полупериметра

$$r = \frac{a + b + c}{2}$$
3. Вычисление периметра

$$P = a + b + c$$
4. Вычисление площади

$$S = \sqrt{r(r - a)(r - b)(r - c)}$$
5. Вывод S, P
6. Конец

Разветвляющиеся алгоритмы

При составлении схем алгоритмов часто возникает необходимость проведения анализа исходных данных или промежуточных результатов вычислений и определения дальнейшего порядка выполнения вычислительного процесса в зависимости от результатов этого анализа.

Алгоритмы, в которых в зависимости от выполнения некоторого логического условия происходит разветвление вычислений по одному из нескольких возможных направлений, называют ***разветвляющимися***.

Подобные алгоритмы предусматривают выбор одного из альтернативных путей продолжения вычислений.

Разветвляющиеся алгоритмы

Каждое возможное направление вычислений называется *ветвью*.

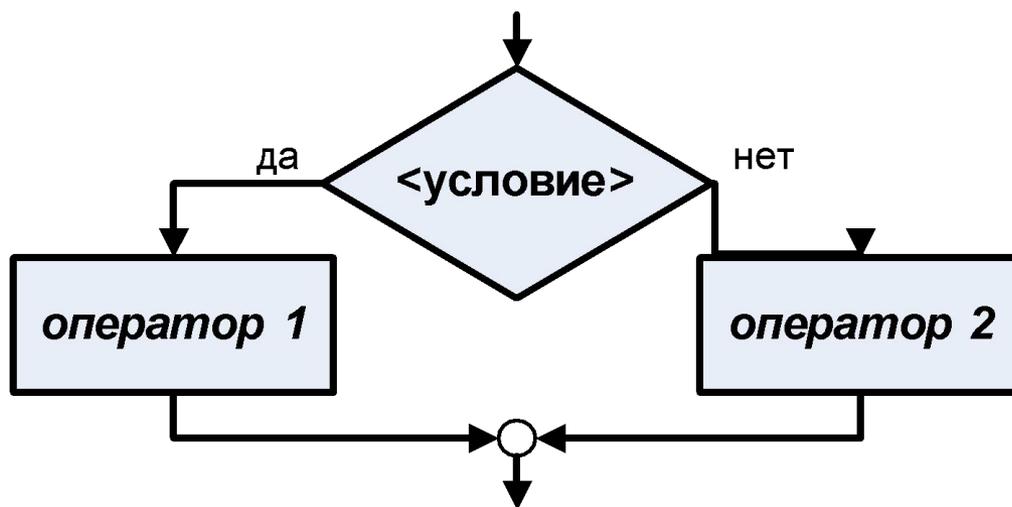
Каждая из ветвей ведет к *общему выходу*, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

Структура **ветвление** существует в трех основных вариантах:

- если–то (неполная форма ветвления);
- если–то–иначе (полная форма ветвления);
- выбор.

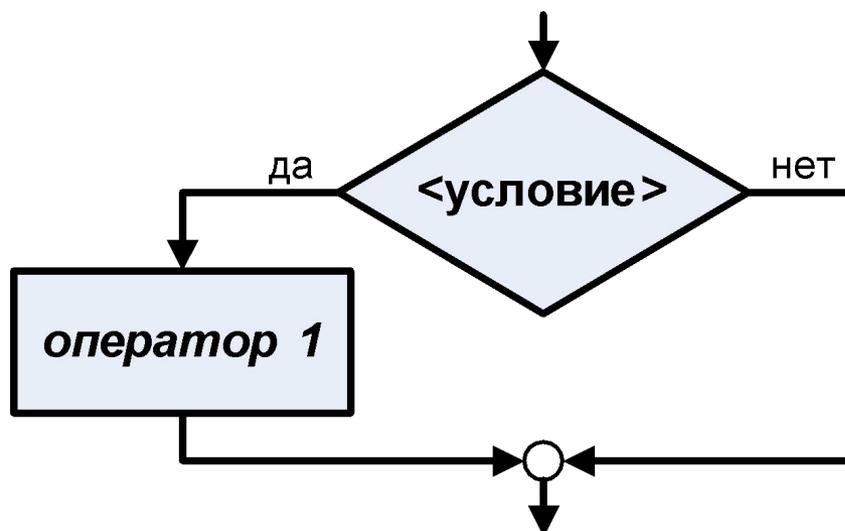
Разветвляющиеся алгоритмы

Полная форма ветвления



Разветвляющиеся алгоритмы

Неполная форма ветвления

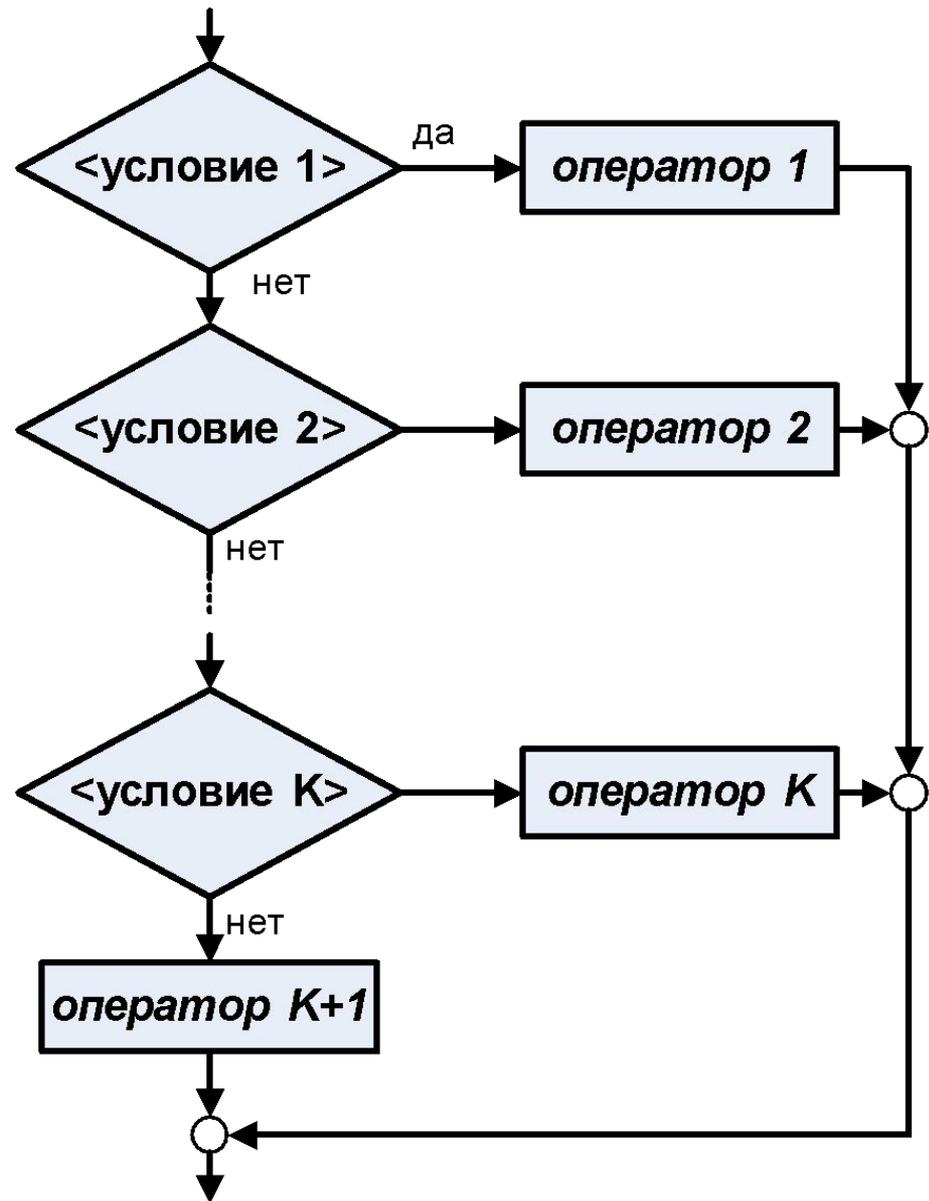


Разветвляющиеся алгоритмы

Выбор

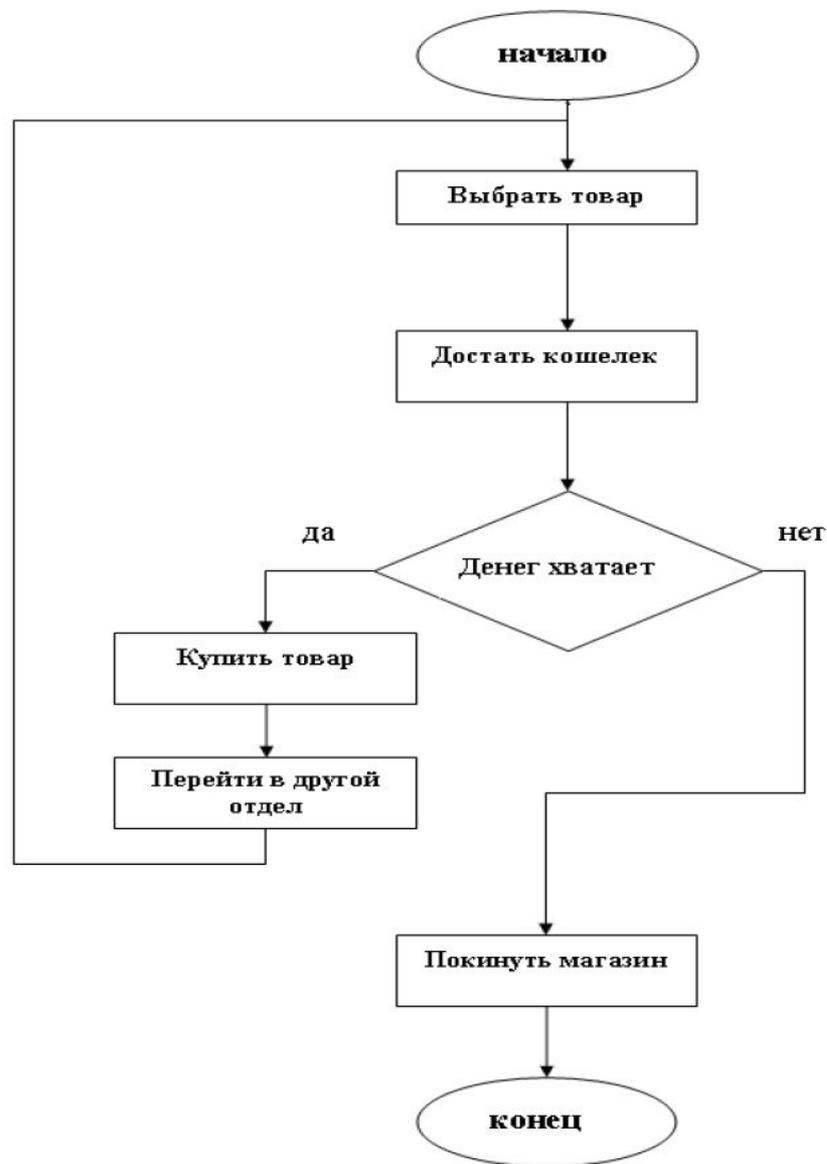
Оператор выбора варианта проверяет поочередно условия и выбирает для исполнения оператор, соответствующий условию, значение которого истинно.

Если все K условий ложны, то выбирается оператор, стоящий по ветке нет последнего условия (последний в списке вариантов).



Разветвляющиеся алгоритмы

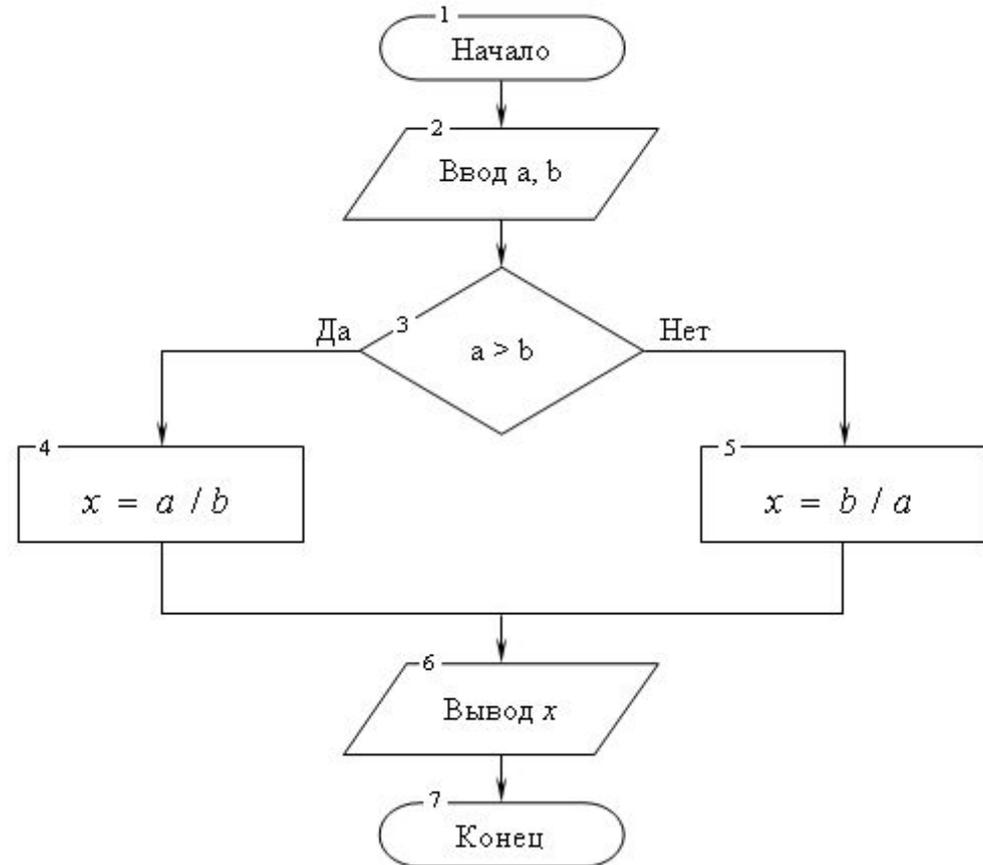
Алгоритм совершения покупок в магазине.



Разветвляющиеся алгоритмы

Даны два числа a и b . Найти

$$x = \begin{cases} a/b, & \text{если } a > b \\ b/a, & \text{если } a \leq b \end{cases}$$



Разветвляющиеся алгоритмы



Циклические алгоритмы

Алгоритм *циклической* структуры предусматривает многократное повторение действий в одной и той же последовательности по одним и тем же математическим зависимостям, но при разных значениях некоторой специально изменяемой величины.

Циклические алгоритмы позволяют существенно сократить объем программы за счет многократного выполнения группы повторяющихся вычислений, так называемого *тела цикла*.

Циклические алгоритмы

Специально изменяемый по заданному закону параметр, входящий в тело цикла, называется ***переменной цикла***.

Переменная цикла используется для подготовки очередного повторения цикла и отслеживания условий его окончания. В качестве переменной цикла используют любые переменные, индексы массивов, аргументы вычисляемых функций и тому подобные величины.

Во время выполнения тела цикла параметры переменной цикла изменяются в интервале от начального до конечного значения с заданным шагом.

Следовательно, при организации циклических вычислений необходимо предусмотреть задание начального значения переменной цикла, закон ее изменения перед каждым новым повторением и ее конечное значение, при достижении которого произойдет завершение цикла.

Циклические алгоритмы

Циклы, в теле которых нет разветвлений и других встроенных в них циклов, называют *простыми*. В противном случае их относят к *сложным*.

Циклические алгоритмы

Существуют следующие разновидности циклических алгоритмов:

Оператор цикла с параметром (типа счетчик);

Оператор цикла с предусловием;

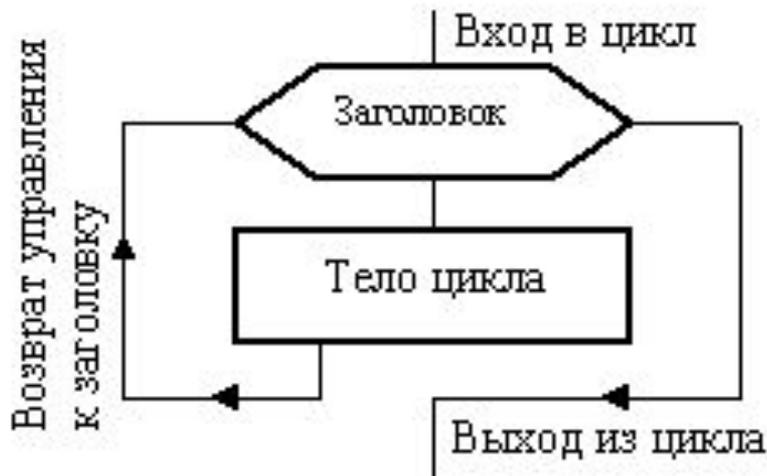
Оператор цикла с постусловием.

Если количество повторов известно заранее, используется оператор цикла типа счетчик, т.е. **детерминированный алгоритм**.

Если количество повторов неизвестно, а задано некоторое условие окончания или продолжения цикла применяются операторы цикла с предусловием или оператор цикла с постусловием. Такие циклы используют для построения **итерационных алгоритмов**.

Цикл типа счетчик

Предписывает выполнять тело цикла **для** всех значений некоторой переменной (параметра цикла) в заданном диапазоне.



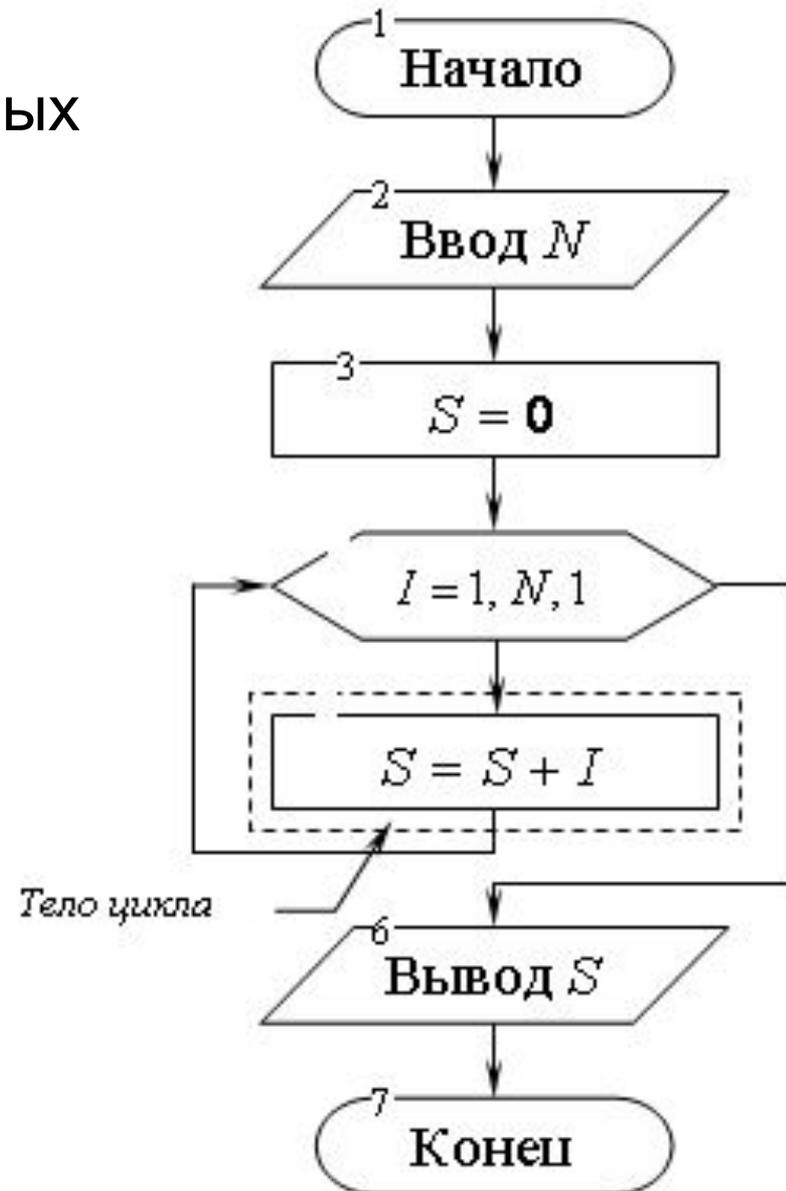
Структура цикла



Структура заголовка цикла

Цикл типа счетчик

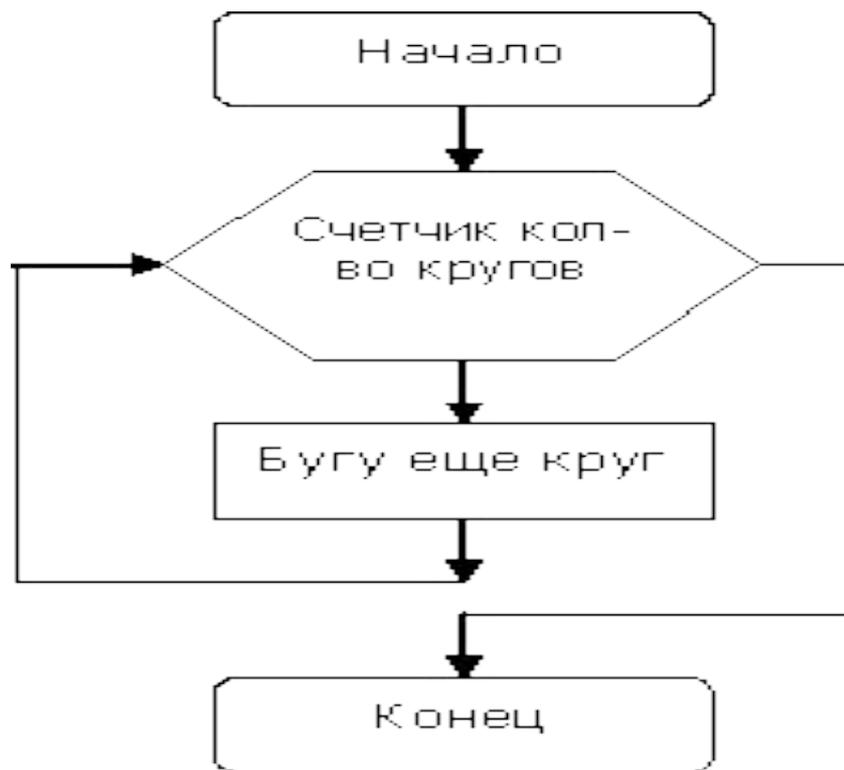
Блок-схема алгоритма нахождения суммы N первых натуральных чисел.



Цикл типа счетчик

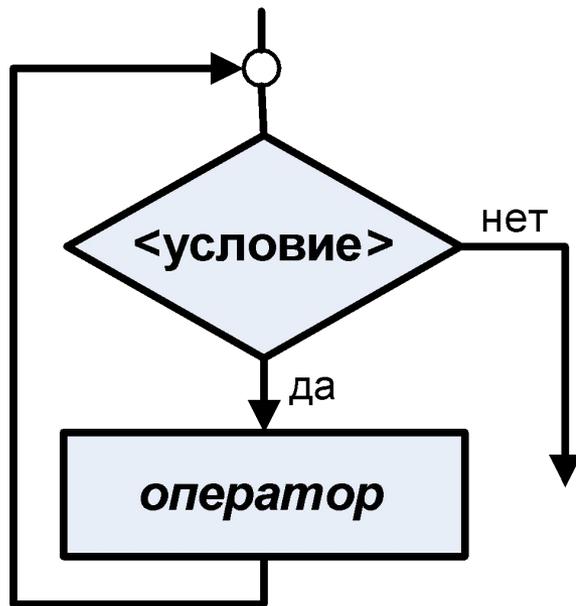
Блок-схема алгоритма

на уроке физкультуры вы должны пробежать некоторое количество кругов вокруг стадиона.



Цикл с предусловием

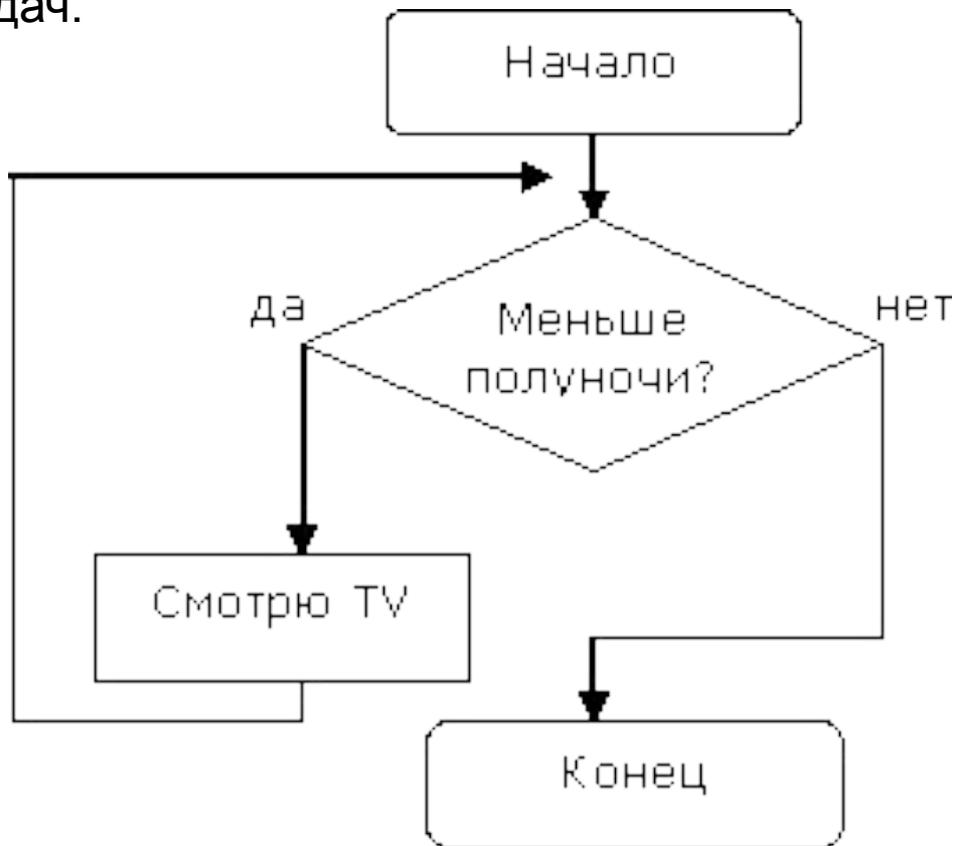
Тело цикла выполняется до тех пор, **пока** условие истинно. Если при первой проверке условие оказалось ложным, оператор не выполняется ни разу.



Перед началом цикла должны быть инициализированы переменные, входящие в условие цикла. Чтобы избежать «зацикливания», среди операторов тела цикла обязательно требуется предусмотреть изменение переменных, входящих в проверяемое условие.

Цикл с условием

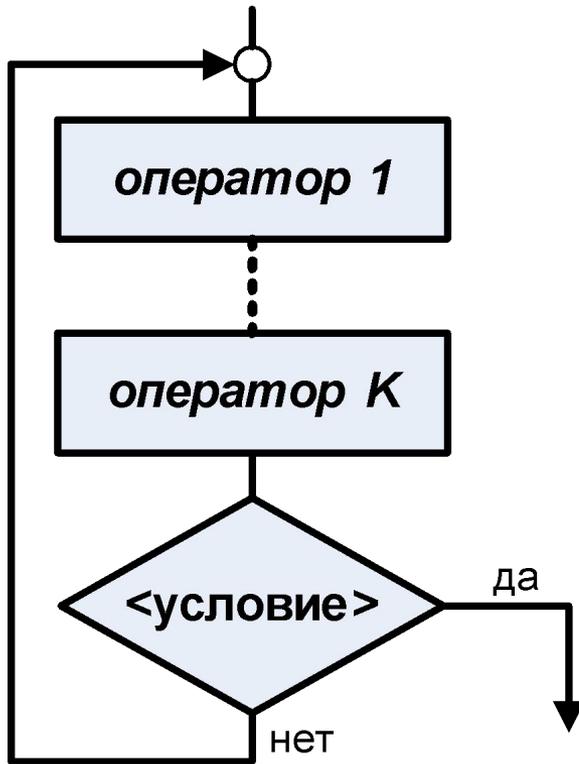
В субботу вечером вы смотрите телевизор. Время от времени поглядываете на часы и если время меньше полуночи, то продолжаете смотреть телевизор, если это не так, то вы прекращаете просмотр телепередач.



Цикл с постусловием

Оператор используется, когда количество повторений заранее неизвестно, а задано некоторое условие выхода из цикла.

Тело цикла выполняется **до тех пор**, пока условие ложно.



Если при первой проверке условие оказалось истинным, оператор цикла выполнится один раз.

Цикл с постусловием

Вам надо поточить все карандаши в коробке.

Вы точите один карандаш и откладываете его в сторону.

Затем проверяете, остались ли карандаши в коробке.

Если условие ложно, то снова выполняется действие 'заточить карандаш'.

Как только условие становится истинным, то цикл прекращается.

