

Блок-схемы

Введение

Алгоритмы и алгоритмизация

Блок-схемы

Элементы блок-схем

Правила составления блок-схем

Примеры составления блок-схем

Пример 1. Определите, является ли число a делителем числа b .

Пример 2. Напишите программу вывода на экран числа, вводимого с клавиатуры. После ввода числа должно следовать сообщение: «That`s the number you entered <number>».

Пример 3. Если число m больше числа n , то выведите на экран «Number $m > n$ », Если число m меньше числа n , то выведите на экран «Number $m < n$ », в противном случае выведите сообщение «The numbers are equal».

Пример 4. Дан радиус окружности. Найдите ее диаметр.

Пример 5. Найдите сумму всех целых чисел от 100 до 500.

Пример 6. Найдите сумму всех целых чисел от a до 500 (значение a вводится с клавиатуры, $a < 500$).

Ссылки на дополнительную информацию

Ссылки на источники

Введение

Это небольшой материал, в котором собрана информация о блок-схемах: что это такое, для чего они нужны и из чего состоят. Так же, в конце будут добавлены ссылки на ресурсы, в которых вы сможете найти дополнительную информацию по данной теме.

Алгоритмы и алгоритмизация

Прежде чем говорить о блок-схемах, давайте разберемся, что же такое алгоритм, ведь с помощью блок-схем мы и будем их описывать.

В интернете можно найти много определений алгоритма:

- Это инструкция о том, в какой последовательности нужно выполнить действия при переработке исходного материала в требуемый результат;
- Это последовательность точных предписаний, понятных исполнителю (компьютеру, роботу и пр.), совершить последовательность действий, направленных на достижение конкретного результата;
- Это набор последовательных инструкций, которые описывают порядок поведения программы для достижения нужной цели;

Так или иначе, все сводится к одному, алгоритм – это последовательность действий приводящих к нужному результату.

Из алгоритма вытекает следующий термин, это алгоритмизация. У нее, так же, как и у алгоритма существует много вариантов определения:

- Это процесс создания некоторых алгоритмов для решения определенного круга задач;
- Это процесс исследования свойств алгоритмов, с целью дальнейшего использования уже существующих видов алгоритмов для решения новых задач;
- Это совокупность приемов и способов составления алгоритмов для решения алгоритмических задач;

Что общего можно вынести из этих определений? Если алгоритм - это набор действий для решения задачи, то алгоритмизация – это процесс создания этого самого алгоритма.

Алгоритмы, в свою очередь, делятся на типы, вот некоторые из них:

- Линейные. Содержат действия, которые идут друг за другом, каждое начинается после того, как закончится предыдущее. Содержат действия, которые будут выполняться много раз в определенном порядке;
- Разветвляющиеся(Ветвящиеся). Содержат действия, которые выполняются, только если верны некоторые условия. Содержат несколько последовательностей действий в зависимости от каких-либо условий;
- Циклические. Содержат действия, которые должны выполняться какое-то количество раз подряд. Содержат действия, которые выполняются в цикле. Цикл может включать в себя одно действие или последовательность, а количество повторений может быть фиксированным или зависеть от условия;
- Вспомогательные. Содержат действия, которые исполняются в другом алгоритме. Содержат действия, которые решают какую-то подзадачу другого алгоритма;

Блок-схемы

Теперь перейдем к блок-схемам, что о них пишут в интернете:

- Это совокупность символов, соответствующих этапам работы алгоритма и соединяющих их линий;
- Это последовательность блоков, предписывающих выполнение определенных операций, и связей между этими блоками;
- Это тип схем описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединённых между собой линиями, указывающими направление последовательности;

Что в итоге? Блок-схема – это схема описывающая какой-либо алгоритм, состоящая из блоков соединенных линиями, в которой каждый блок является каким-либо действием.

Каждый из блоков отвечает за выполнение одного, либо нескольких действий. Так как возможны разные действия, то каждому типу действия соответствует геометрическая фигура.

Примечание

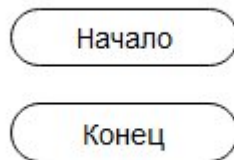
Существует специальный ГОСТ 19.701-90, в котором определены правила создания блок-схем, вы всегда можете найти и ознакомиться с ним в Интернете.

Примечание

Не стоит забывать о словесном описании алгоритмов. Данная практика довольно эффективна на начальных этапах разработки, когда вы сначала описываете словами каждое действие, которое необходимо выполнить для достижения результата.

Элементы блок-схем

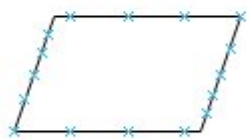
Блок начала/конца(Определяют начало и конец работы алгоритма)



Функциональный блок(Выполнение операций над данными)



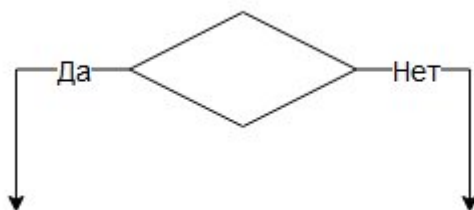
Блок ввода/вывода(Операции ввода и вывода данных)



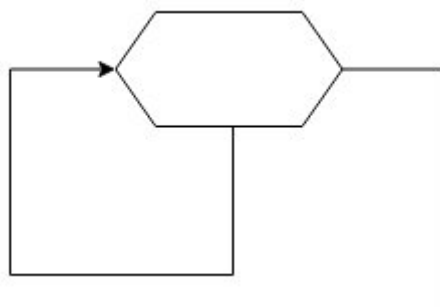
Блок подпрограммы(Вызовы внешних функций)



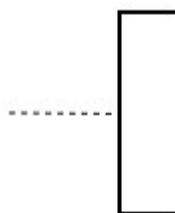
Блок условия/ветвления(Содержит условие)



Блок цикла



Блок комментария



Правила составления блок-схем

Правил не много, давайте выделим основные:

1. Каждая блок-схема должна иметь блок «Начало» и блок «Конец»;
2. «Начало» должно быть соединено с блоком «Конец» линиями по каждой из имеющихся на блок-схеме ветвей;
3. В блок-схеме не должно быть блоков, кроме блока «Конец», из которых не выходит линия потока, равно как и блоков, из которых управление передается «в никуда»;
4. По отношению к блокам линии могут быть входящими и исходящими. Одна и та же линия потока является исходящей для одного блока и входящей для другого;
5. От блока «Начало» в отличие от всех остальных блоков линия потока только выходит, так как этот блок – первый в блок-схеме;
6. Блок «Конец» имеет только вход, так как это последний блок в блок-схеме;

Примечание

Как правило, действия в блок-схеме идут сверху вниз, так она получается более удобной для чтения/понимания.

Примечание

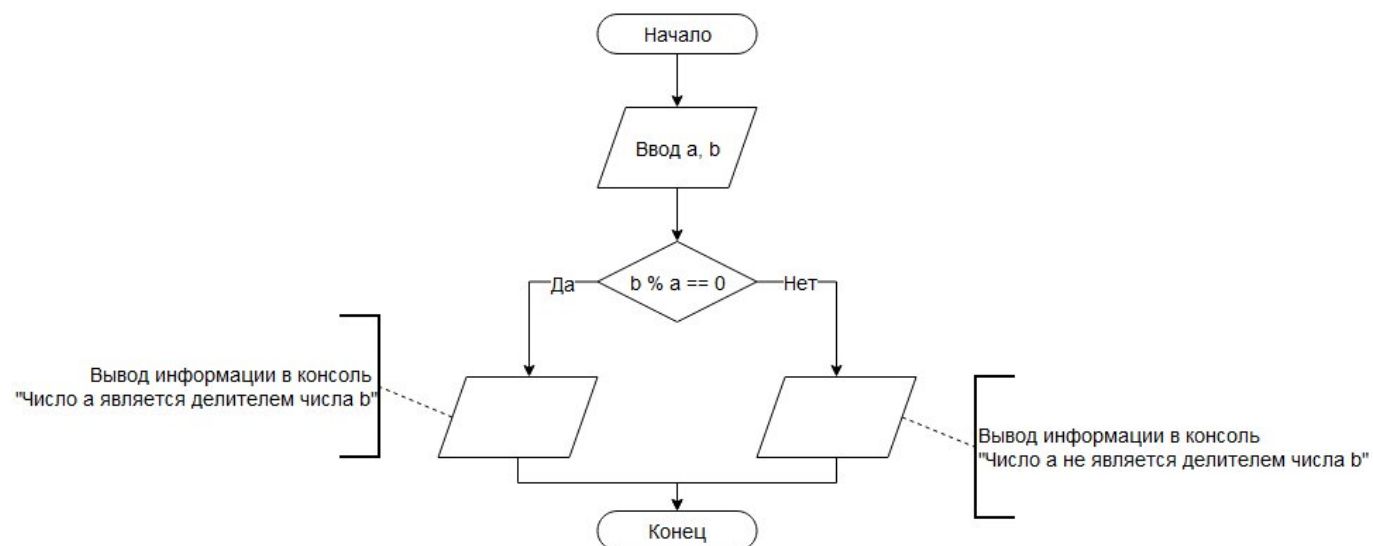
Размер блоков «Начало» и «Конец» должен быть размерами примерно 1см на 4см, размер остальных блоков примерно 2см на 4см. На сайте draw.io это примерно 28pt на 113pt и 57pt на 113pt.

Примечание

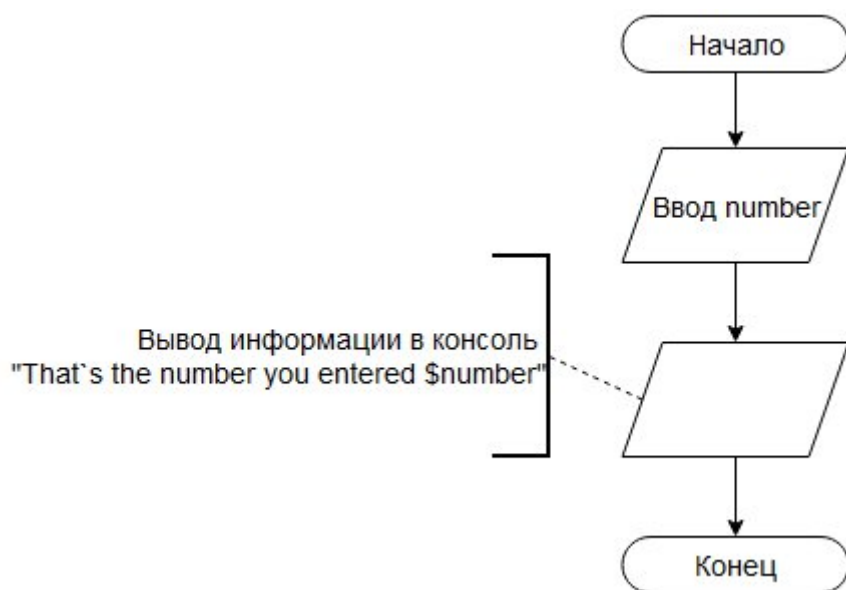
Не стоит изменять размеры блоков. Если текст не влезает в блок, оставьте блок пустым и напишите в комментарии то, что нужно.

Примеры составления блок-схем

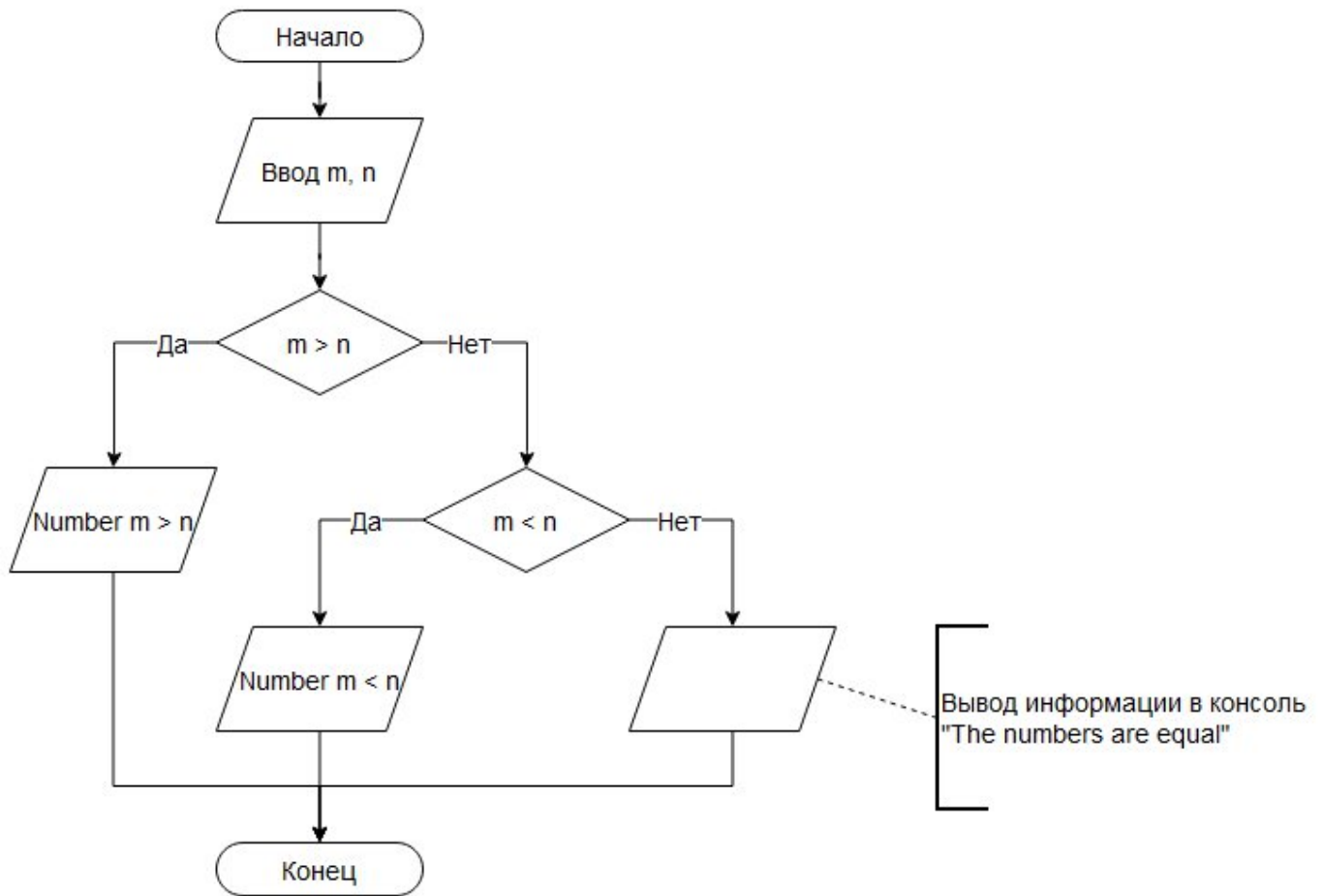
Пример 1. Определите, является ли число a делителем числа b .



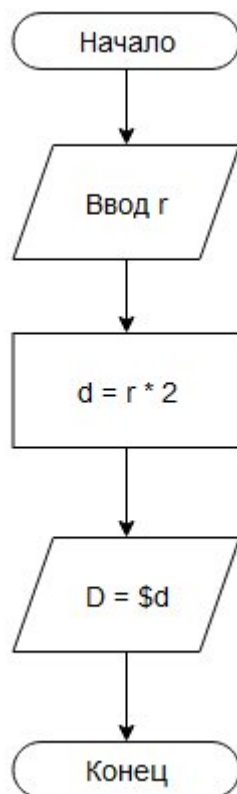
Пример 2. Напишите программу вывода на экран числа, вводимого с клавиатуры. После ввода числа должно следовать сообщение: «That`s the number you entered <number>».



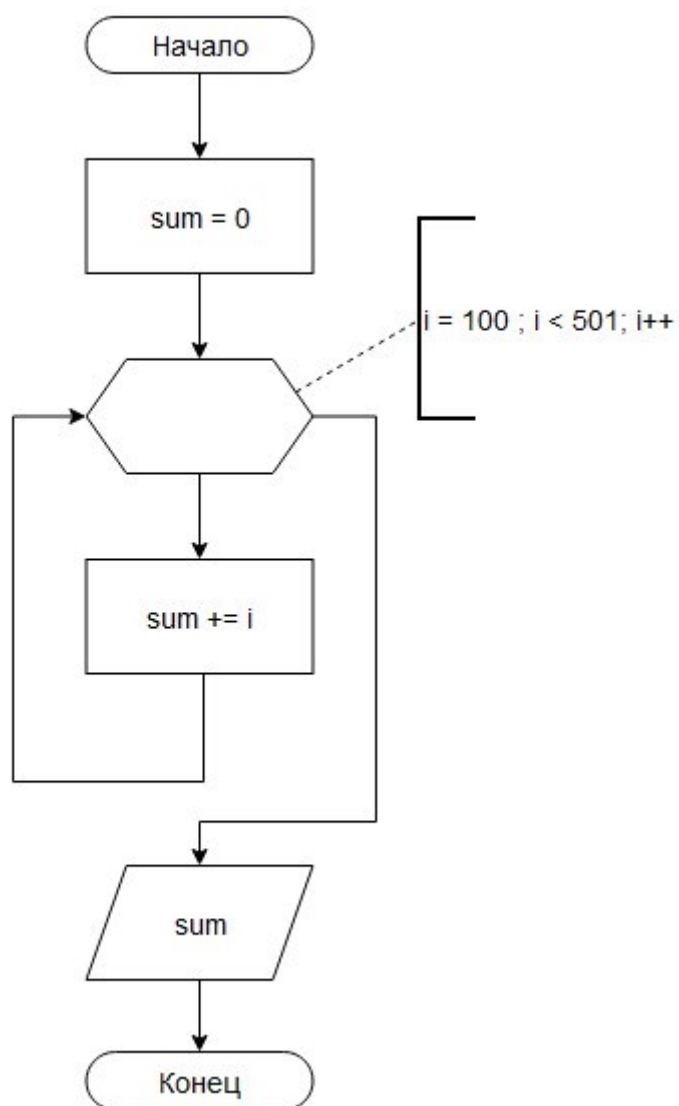
Пример 3. Если число m больше числа n , то выведите на экран «Number $m > n$ », Если число m меньше числа n , то выведите на экран «Number $m < n$ », в противном случае выведите сообщение «The numbers are equal».



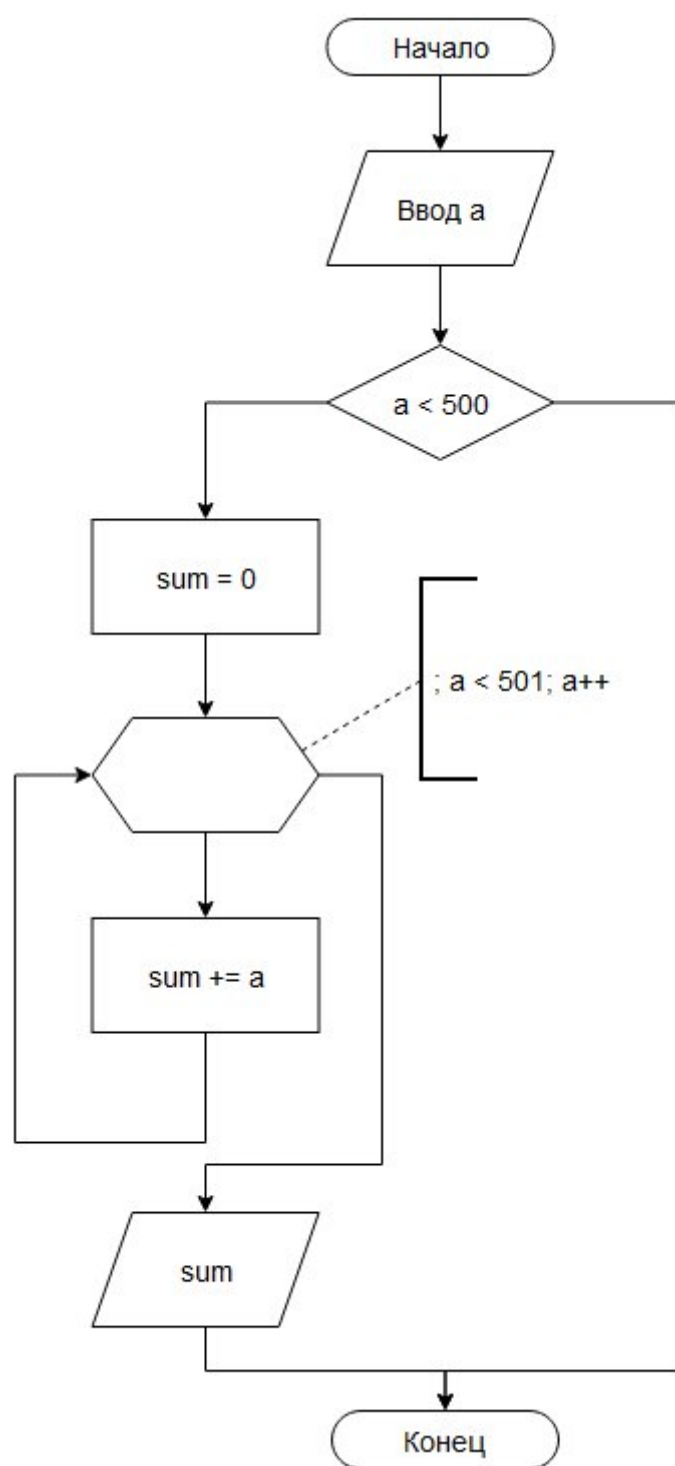
Пример 4. Дан радиус окружности. Найдите ее диаметр.



Пример 5. Найдите сумму всех целых чисел от 100 до 500.



Пример 6. Найдите сумму всех целых чисел от a до 500 (значение a вводится с клавиатуры, $a < 500$).



Ссылки на дополнительную информацию

- [Как улучшить блок-схемы алгоритмов по ГОСТ 19.701-90? Эргономичный визуальный алгоритмический язык Дракон](#)

Примечание

Для создания блок-схем, и не только, отлично подходит сайт draw.io, хотя это дело вкуса:)

Ссылки на источники

- [Блок-схема](#)
- [Блок-схемы алгоритмов. Назначение блоков данных](#)
- [Справочное руководство по составлению алгоритмов](#)
- [Блок-схемы алгоритмов. ГОСТ. Примеры](#)
- [Представление алгоритмов в виде блок-схем](#)