

# Что такое алгоритм?

**Алгоритм** — это точное описание порядка действий, которые должен выполнить исполнитель для решения задачи за конечное время.

**Исполнитель** — это устройство или одушевленное существо (человек), способное понять и выполнить команды, составляющие алгоритм.

## Свойства алгоритма

**Дискретность** — алгоритм состоит из отдельных команд, каждая из которых выполняется за конечное время.

**Детерминированность** (определённость) — при каждом запуске алгоритма с одними и теми же исходными данными получается один и тот же результат.

**Понятность** — алгоритм содержит только команды, входящие в систему команд исполнителя.

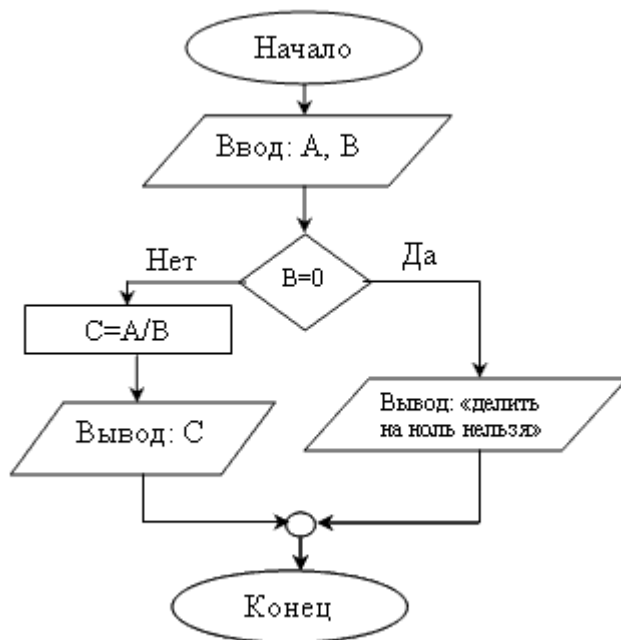
**Конечность (результативность)** — для корректного набора данных алгоритм должен завершаться через конечное время.

**Корректность** — для допустимых исходных данных алгоритм должен приводить к правильному результату.

**Массовость** — алгоритм можно использовать для разных исходных данных.

## Представление алгоритма

1. В графическом виде (блок-схема)



2. В виде псевдокода.

```

ввести два числа
если знаменатель равен нулю, то вывести "делить на ноль нельзя"
иначе, вычислить их частное
вывести частное на экран
  
```

Псевдокод – алгоритм на русском языке с элементами языка программирования.  
Компьютер не может исполнять псевдокод.

3. В виде программного кода

## Простейшая программа

```

In [3]: # Это пустая программа
# Всё что пишется в строке после решетки – комментарий
# Комментарии не обрабатываются компьютером.
# Они нужны для того, что делать код более понятным
  
```

```

In [ ]: """
Комментарии
могут
быть
многострочными
в тройных кавычках
"""
  
```

## Вывод на экран

Функция `print()` позволяет выводить информацию на экран. Необходимая информация пишется внутри скобок `()`

```
In [7]: print() #Вывод пустой строки
```

```
In [8]: print(4) #Вывод числа
```

4

```
In [9]: print ( "Hello world" ) #Вывод строки из символов
```

Hello world

```
In [10]: #Пример  
print ( "2+2=?" )  
print ( "Ответ: 4" )
```

2+2=?

Ответ: 4

## Чувствительность к отступам

```
In [ ]: #Все команды должны начинаться с начала строки  
print() #правильно  
    print() #не правильно  
print() #правильно
```

## Дополнительные символы

```
In [13]: print("Первая\nВторая") #Символ "\n" добавляет переход на новую строку
```

Первая

Вторая

```
In [14]: print("До\tПосле") #Символ "\t" добавляет табуляционный отступ
```

До

После