

# **Основы алгоритмизации и программирования**

## **Алгоритмы**

# Очень часто в жизни человек действует по какому-то определенному плану.

## Пример: нужно пришить пуговицу

1. Взять катушку ниток
2. Отмотать нить нужной длины
3. Отрезать нить от катушки
4. Вдеть нить в иголку
5. Завязать на конце нити узелок
6. Продеть иглу через ткань (с изнаночной стороны) и отверстие пуговицы
7. Продеть иглу через другое отверстие пуговицы и ткань.
8. Повторить п.п. 6 и 7 пять раз
9. Отрезать нить
10. Положить катушку и иглу на место



# Алгоритм

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

Появление алгоритмов связывают с зарождением математики. Более 1000 лет назад (в 825 году) ученый из города Хорезма **Абдулла Мухаммед бен Мусса аль-Хорезми** создал книгу по математике, в которой описал способы выполнения арифметических действий над многозначными числами.

Само слово **алгоритм** возникло в Европе после перевода на латынь книги этого математика.



# Алгоритм открывания замка двери

1. Достать ключ из кармана.
2. Вставить ключ в замочную скважину.
3. Повернуть ключ два раза по часовой стрелки.
4. Откинуть дужку.
5. Замок открыт.



# Свойства алгоритмов

1. **Дискретность** (алгоритм должен состоять из конкретных действий, следующих в определенном порядке);
2. **Детерминированность** (любое действие должно быть строго и недвусмысленно определено в каждом случае);
3. **Конечность** (каждое действие и алгоритм в целом должны иметь возможность завершения);
4. **Массовость** (один и тот же алгоритм можно использовать с разными исходными данными);
5. **Результативность** (отсутствие ошибок, алгоритм должен приводить к правильному результату для всех допустимых входных значениях).

# Алгоритм открывания замка двери

1. Достать ключ из кармана.
2. Вставить ключ в замочную скважину.
3. Повернуть ключ два раза по часовой стрелки.
4. Откинуть дужку.
5. Замок открыт.



# Формы представления алгоритма

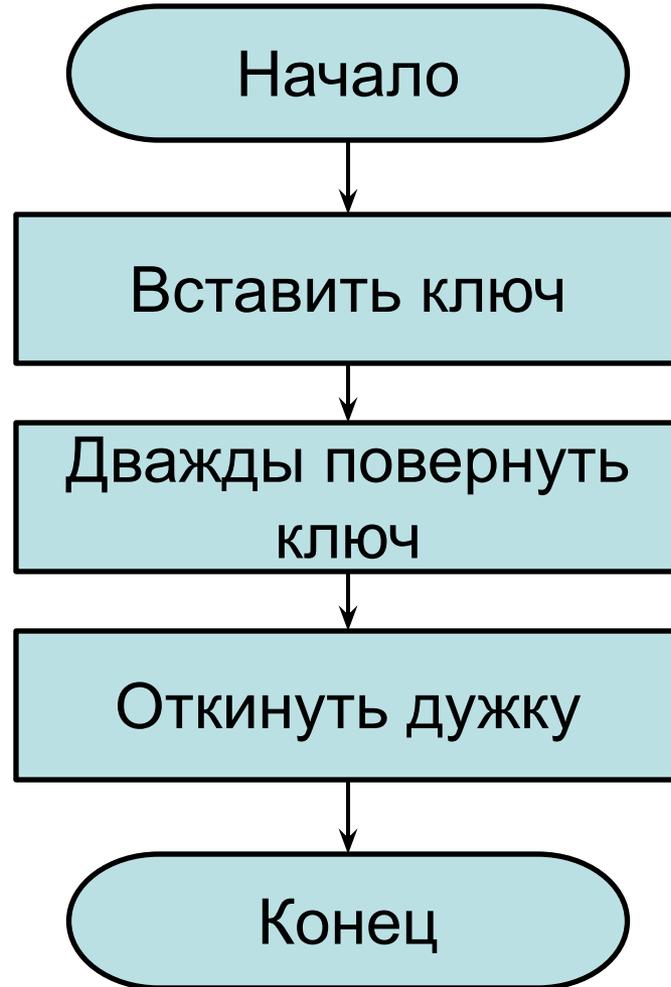
- В устной форме.
- В письменной форме на естественном языке.
- В письменной форме на формальном языке.
- В графической форме – в виде блок-схемы, которая составляется из стандартных графических объектов.

# Письменная форма

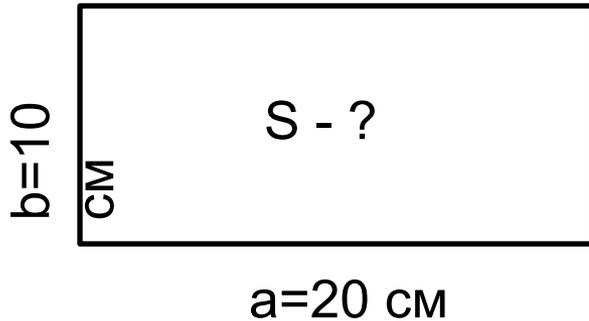
1. Достать ключ из кармана.
2. Вставить ключ в замочную скважину.
3. Повернуть ключ два раза по часовой стрелки.
4. Откинуть дужку.
5. Замок открыт.



# Графическая форма (блок-схема)



# Пример записей алгоритма



**Дано:**

$$a=20 \text{ см}$$

$$b=10 \text{ см}$$

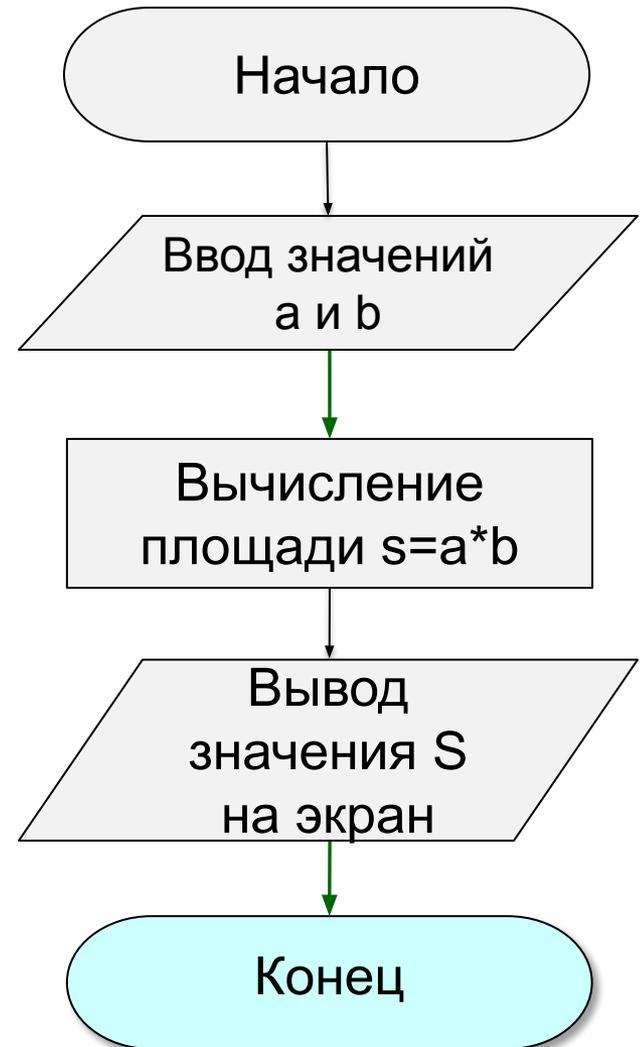
**Решение:**

$$S=a*b$$

$$S=20*10=200$$
$$см^2$$

**Ответ:**

$$S=200 \text{ см}^2$$



# Стадии создания алгоритма

1. Алгоритм должен быть представлен в форме, понятной **человеку**, который его **разрабатывает**.
2. Алгоритм должен быть представлен в форме, понятной тому **объекту**, который **будет его выполнять**.

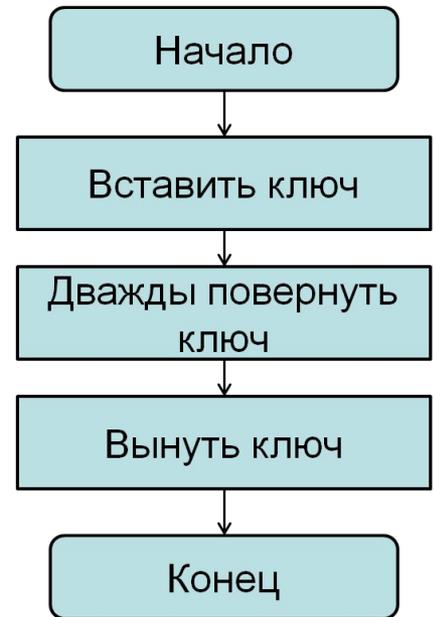
# Стадии создания алгоритма

1. Достать ключ из кармана.
2. Вставить ключ в замочную скважину.
3. Повернуть ключ два раза против часовой стрелки.
4. Вынуть ключ.
5. Замок открыт.

```
program pifagor;  
uses GraphABC;  
begin  
  SetWindowSize(300,300);  
  SetWindowTitle('Сложение чисел');  
  SetColor(15);  
end;
```

```
00110000 01111100 11101001 00110110 01001101 00100101 11100011  
11001110 00010001 10111111 11000001 00001000 00000000 00101011  
11100001 00000011 00011000 01111101 11101001 00110110 01001101  
00100101 11100011 11001110 00010001 10111111 11000001 00001000  
00000000 00101011 11100001 00000011 00011000 10000100 01111000  
10000000 01101101 00100001 01111101 11001111 00010001 10000000  
00011100 00001000 00000000 00101011 11100001 00000011 00011000  
01111110 11101001 00110110 01001101 00100101 11100011 11001110  
00010001 10111111 11000001 00001000 00000000 00101011 11100001  
00000011 00111000 01100000 10011110 11111100 00110110 01100101
```

end;



# Исполнители алгоритмов

- **Исполнитель** - объект, который выполняет алгоритм.
- **Компьютер** - автоматический исполнитель алгоритмов.
- Команды, которые понимает и может выполнить исполнитель, образуют **систему команд исполнителя**.
- Алгоритм, записанный на «понятном» компьютеру языке, называется **программой**.