

# Алгоритмы

Введение в программирование

A series of horizontal lines of varying lengths and colors (teal, light blue, white) extending from the right side of the text area across the top of the slide.

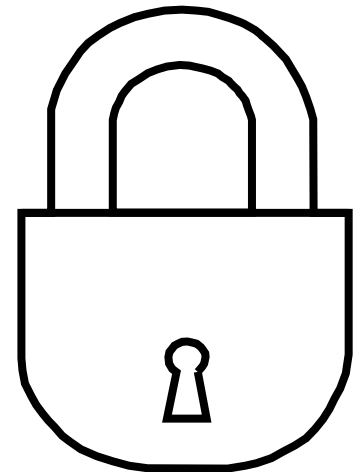
# Алгоритм

Появление алгоритмов связывают с зарождением математики. Более 1000 лет назад (в 825 году) ученый из города Хорезма Абдулла (или Абу Джафар) Мухаммед бен Муса 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 - описание последовательности действий (план), строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов.

# Алгоритм открывания двери

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



# Свойства алгоритмов

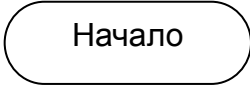
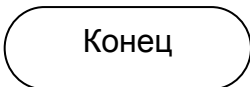
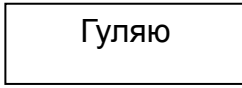
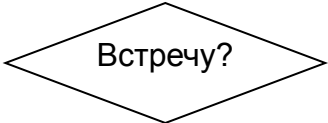
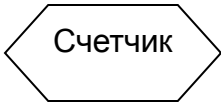
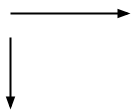
1. **Дискретность** (алгоритм должен состоять из конкретных действий, следующих в определенном порядке);
2. **Детерминированность** (любое действие должно быть строго и недвусмысленно определено в каждом случае);
3. **Конечность** (каждое действие и алгоритм в целом должны иметь возможность завершения);
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).

# Виды алгоритмов

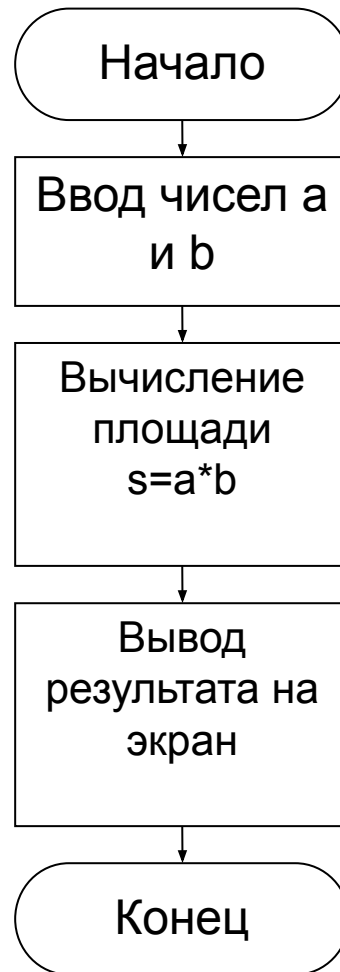
1. **Линейный алгоритм** (описание действий, которые выполняются однократно в заданном порядке);
2. **Циклический алгоритм** (описание действий, которые должны повторяться указанное число раз или пока не выполнено задание);
3. **Разветвляющийся алгоритм** (алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий)
4. **Вспомогательный алгоритм** (алгоритм, который можно использовать в других алгоритмах, указав только его имя).

# Представления алгоритма

- В устной форме.
- В письменной форме на естественном языке.
- В письменной форме на формальном языке.
- Для более наглядного представления алгоритма широко используется графическая форма - **блок-схема**, которая составляется из стандартных графических объектов.

Вид стандартного графического объекта	Назначение
	Начало алгоритма
	Конец алгоритма
	Выполняемое действие записывается внутри прямоугольника
	Условие выполнения действий записывается внутри ромба
	Счетчик кол-во повторов
	Последовательность выполнения действий

# Пример записи алгоритма в виде блок-схемы





# 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 (в том числе и человеку), который будет выполнять описанные в алгоритме действия.

# Исполнители алгоритмов

Объект, который будет выполнять алгоритм, обычно называют исполнителем.

- Исполнитель - объект, который выполняет алгоритм.

Идеальными исполнителями являются машины, роботы, компьютеры...

- Компьютер – автоматический исполнитель алгоритмов.
- Алгоритм, записанный на «понятном» компьютеру языке программирования, называется программой.

# Вопросы:

- Что такое алгоритм? Приведите примеры алгоритмов.
- Какие свойства алгоритмов вы знаете?
- Какие виды алгоритмов вы знаете?
- Какие способы записи алгоритмов вы знаете?
- Что такое исполнитель алгоритмов?
- Что такое программа?

```
PRINT "Привет!"
```

CLS

PRINT "Привет!"

```
CLS
```

```
a=3
```

```
b=4
```

```
s=a*b
```

```
PRINT "Площадь прямоугольника  
равна ",s
```

```
END
```

**CLS**

**INPUT "Введите длину: ", a**

**INPUT "Введите ширину: ", b**

**s=a\*b**

**PRINT "Площадь прямоугольника  
равна ",s**

**END**