

# Алгоритмизация

# Что такое алгоритм?

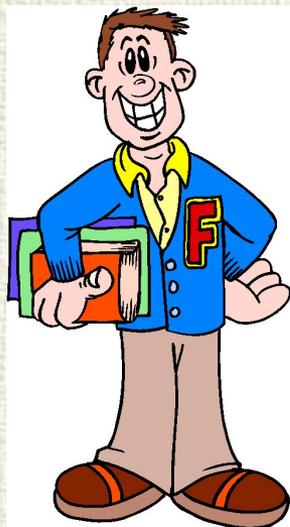
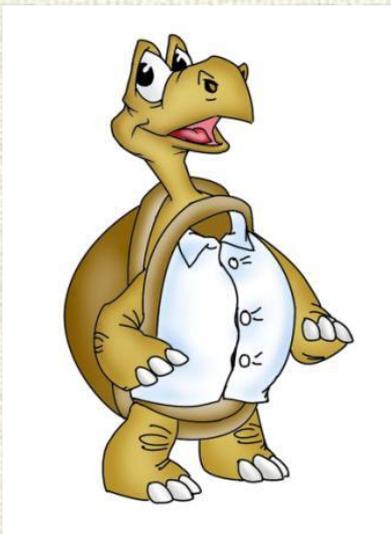


Абу Абдуллах Мухаммед  
ибн Муса аль-Хорезми

Алгоритм — точная конечная система правил (последовательность действий), предназначенная для исполнителя и приводящая к определенному результату.

# Исполнители

**Исполнитель** – человек, животное или машина, способные  
понимать и выполнять некоторые команды.



**Среда исполнителя** – предметы, которые окружают исполнителя и с которыми он работает.

**Система Команд Исполнителя (СКИ)** – набор команд, понятных исполнителю. Исполнитель может выполнить только те команды, которые входят в его СКИ.

# Свойства алгоритмов

**Дискретность** — алгоритм должен быть разбит на конечное число шагов, выполняемых последовательно.

**Понятность** — алгоритм должен быть понятен конкретному исполнителю.

**Массовость** — однажды составленный алгоритм должен быть применим для решения сходных задач с разными исходными данными.

**Детерминированность (определенность)** — каждая команда алгоритма должна быть однозначно определена.

**Результативность** — выполнение алгоритма приводит к результату.

# Типы алгоритмов

## 1) Линейный



## 2) Ветвление

## 3) Циклический



# Этапы решения задач на ПК

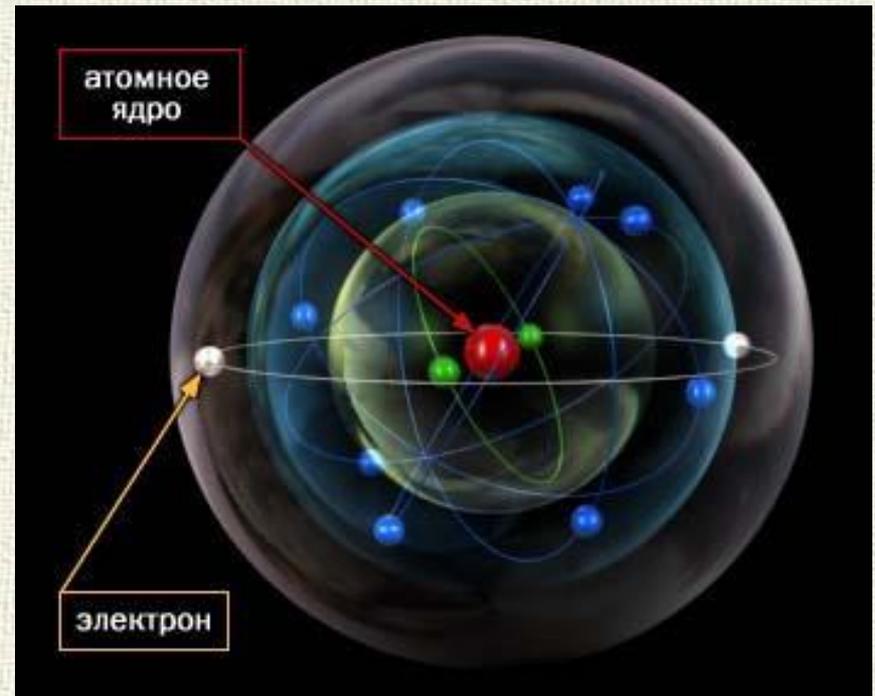
## 1) Постановка цели



## 2) Создание сценария

# Этапы решения задач на ПК

3) Логическое и математическое моделирование.



Модель — упрощенная копия объекта, отражающая его изучаемые стороны.

# Этапы решения задач на ПК

Арг – описание входных данных

$a$  – {длина прямоугольника, см}

$b$  – {ширина прямоугольника, см}

Рез – описание выходных данных

$S$  – {площадь прямоугольника, см<sup>2</sup>}

Связь – получение результатов из аргументов при помощи действий

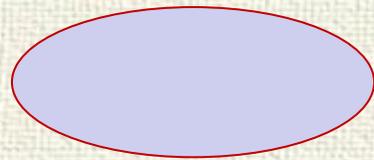
Ввод ( $a, b$ );

$S = a * b$ ;

Вывод ( $S$ );

# Этапы решения задач на ПК

## 4) Построение блок-схемы.



начало,  
конец



СВЯЗЬ



ВВОД,  
ВЫВОД



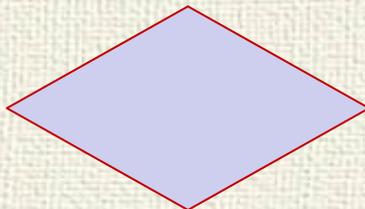
ЦИКЛ С  
параметром



действие



подпрограммы



условие



# Этапы решения задач на ПК

## 6) Тестирование и отладка



# Языки программирования

- 1) Низкоуровневые
  - Машинный код;
  - Ассемблеры.
- 2) Высокоуровневые
  - Процедурные (Basic, Си, Pascal).
  - Объектно-ориентированные (C++, Java, Python, Ruby)
  - Скриптовые (Perl, php, Python)
  - Учебные (Logo, Scratch, Pascal)

## Типы трансляторов

Компиляторы - переводят целиком всю программу, с ЯП на машинный язык, после чего программа записывается в память компьютера и лишь потом выполняется;

Интерпретаторы - переводят команды входной программы по очереди и немедленно выполняют их.