

# \*АЛГОРИТМЫ

Свойства алгоритмов.

Способы записи алгоритмов

# Происхождение и развитие понятия алгоритма

**Алгоритм** – происходит от Al Horithmi – латинского написания арабского имени среднеазиатского математика IX века аль - Хорезми

IX век



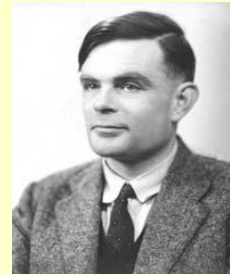
787 – 850 г.

XX век

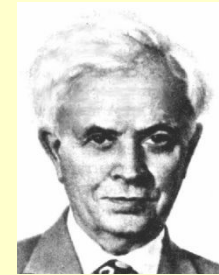
Возникает научное направление  
**ТЕОРИЯ АЛГОРИТМОВ**  
Направление исследований:  
разработка универсальной алгоритмической модели

30 – е годы

Основатели теории алгоритмов



1912 - 1954  
г.



1903 - 1979  
г.



С появлением ЭВМ (2-я половина XX века) понятие **АЛГОРИТМА** связывается с **ПРОГРАММИРОВАНИЕМ**. Появляется большое количество алгоритмических языков: Фортран, Паскаль, Бейсик . . .

# Происхождение и развитие понятия алгоритма

**Алгоритм** – происходит от Al Horithmi – латинского написания арабского имени среднеазиатского математика IX века аль - Хорезми

IX век

XX век

Основатели теории алгоритмов



787 – 850 г.

Возникает научное направление  
**ТЕОРИЯ АЛГОРИТМОВ**  
Направление исследований:  
разработка универсальной алгоритмической модели



1912 - 1954  
г.



1903 - 1979  
г.

30 – е годы

В XII веке в Европе вышел латинский перевод математического трактата **аль – Хорезми**. Алгоритмами называли описанные в трактате правила выполнения арифметических вычислений в позиционной десятичной системе счисления.

*В наше время понятие алгоритма понимается шире, не ограничиваясь только арифметическими вычислениями.*

# Происхождение и развитие понятия алгоритма

**Алгоритм** – происходит от Al Horithmi – латинского написания арабского имени среднеазиатского математика IX века аль - Хорезми

IX век



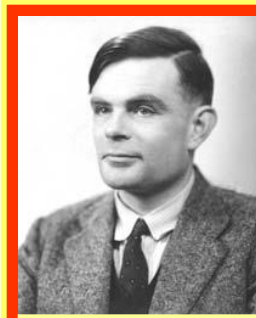
787 – 850 г.

XX век

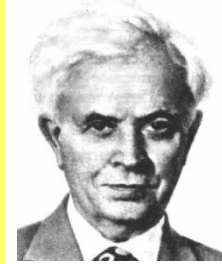
Возникает научное направление  
**ТЕОРИЯ АЛГОРИТМОВ**  
Направление исследований:  
разработка универсальной алгоритмической модели

30 – е годы

Основатели теории алгоритмов



1912 - 1954  
г.



1903 - 1979  
г.

Английский математик **Алан Тьюринг** в 1935 – 1936 годах создает теорию «логических вычисляющих машин». Разработанная им «**Машина Тьюринга**» стала обязательной частью обучения будущих математиков и компьютерщиков. На одной из лондонских гостиниц мемориальная доска гласит: «Здесь родился Алан Тьюринг (1912 – 1954), взломщик кодов и пионер информатики».

# Происхождение и развитие понятия алгоритма

**Алгоритм** – происходит от Al Horithmi – латинского написания арабского имени среднеазиатского математика IX века аль - Хорезми

IX век



787 – 850 г.

XX век

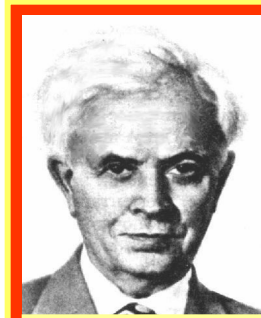
Возникает научное направление  
**ТЕОРИЯ АЛГОРИТМОВ**  
Направление исследований:  
разработка универсальной алгоритмической модели

30 – е годы

Основатели теории алгоритмов



1912 - 1954  
г.



1903 - 1979  
г.

Русский математик **Андрей Марков** в 1947 году ввел понятие «**нормального алгоритма**» и впервые систематически и строго построил общую теорию алгоритмов. Современные языки символьной обработки (Пролог) берут свое начало от нормальных алгоритмов Маркова.

# \*Что такое алгоритм?

**Алгоритм** – понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение определенной цели или на решение поставленной задачи.

***Задание.*** Вы захотели выпить чашечку чаю. Запишите порядок своих действий.

1. Налить в чайник воду.
2. Зажечь газовую горелку.
3. Поставить на нее чайник.
4. Подождать пока вода в чайнике закипит.
5. Отключить газ.
6. В заварной чайник насыпать 2-3 чайные ложки заварки.
7. Залить кипятком и дать настояться 5 минут.
8. Налить чай в чашки.
9. Добавить сахар/молоко/мёд по вкусу.

# \* Свойства алгоритма

**Понятность** – каждый шаг алгоритма должен быть понятен исполнителю;

**Дискретность** (прерывность, отдельность) – алгоритм может быть разбит на шаги;

**Конечность** - выполняемый алгоритм должен приводить к результату за конечное число шагов;

**Результативность** - алгоритм должен быть направлен на получение результата за конечное число шагов;

**Массовость** – алгоритм может быть использован для решения однотипных задач разной направленности.

**Формальность** – возможность выполнять команды механически. Это свойство позволяет поручить исполнение алгоритмов роботам, компьютерам и другим устройствам, т.е. выполнение алгоритма без понимания цели.



# \* Способы записи алгоритмов:

Алгоритмы можно записывать разными способами, называемыми *формой представления алгоритма*.

На практике наиболее распространены следующие формы представления алгоритмов:

- **словесная** (записи на естественном языке);
- **графическая** (стрелки, изображения, блок-схемы);
- **псевдокоды** (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- **программная** (тексты на языках программирования).



# \* Словесная форма

Обычно используется для алгоритмов, ориентированных на исполнителя-человека. Команды такого алгоритма выполняются в естественной последовательности, если не оговорено противного.

## Примеры записи алгоритмов на естественном языке.

### Алгоритм «Съешь конфету»

1. Возьми конфету из вазы.
2. Разверни фантик.
3. Съешь конфету.
4. Фантик выбрось в мусорное ведро.

### Алгоритм «Рисунок»

1. Возьми карандаш.
- «Если ты любишь рисовать, то нарисуй яблоко, иначе напиши, чем ты любишь заниматься».


### Алгоритм «Набери в лесу грибов»

1. Возьми пустую корзину.
2. Прийди в лес.
3. Если нашел съедобный гриб, то положи в корзину.
4. Если корзина еще не полная, то повтори п.3, иначе перейди к п.5.
5. Приди домой.
6. Поставь корзину с грибами на место.

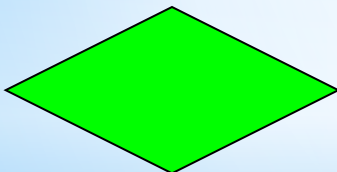
# \* Графическая форма


Шаги алгоритмов обозначаются геометрическими фигурами.

 → Начало или конец

 → Ввод или вывод

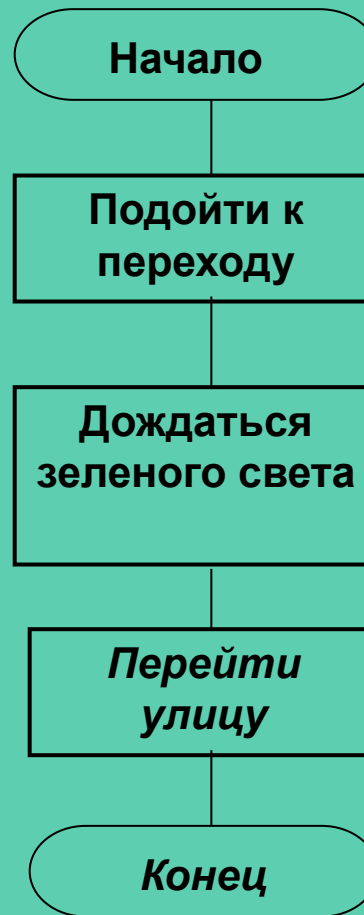
 → Выполнение действия

 → Принятие решения

 → Цикл со счетчиком

 → Переход

**Пример алгоритма  
представленного в форме блок-схемы**



# \* Псевдокод

\* Представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов. Он занимает промежуточное место между естественным и формальным языком.

АЛГ <Имя алгоритма>

НАЧ

Ввод <Исходные данные>

<Серия команд>

Вывод <Результат>

КОН

АЛГ Вычислить  $Y=R+T-X$

НАЧ

Ввод R, T, X

A1:= R+T

Y:= A1-X

Вывод Y

КОН

Примеры записи алгоритмов на алгоритмическом языке.

АЛГ Дождь

НАЧ

Подойди к окну

ЕСЛИ Идет дождь

ТО Остайся дома

ИНАЧЕ Иди гулять

ВСЕ

КОН



# \* Программа

Program ostatok;

Uses crt;

Var a, b, max: real;

Begin

ClrScr;

Readln (a, b);

If a>b

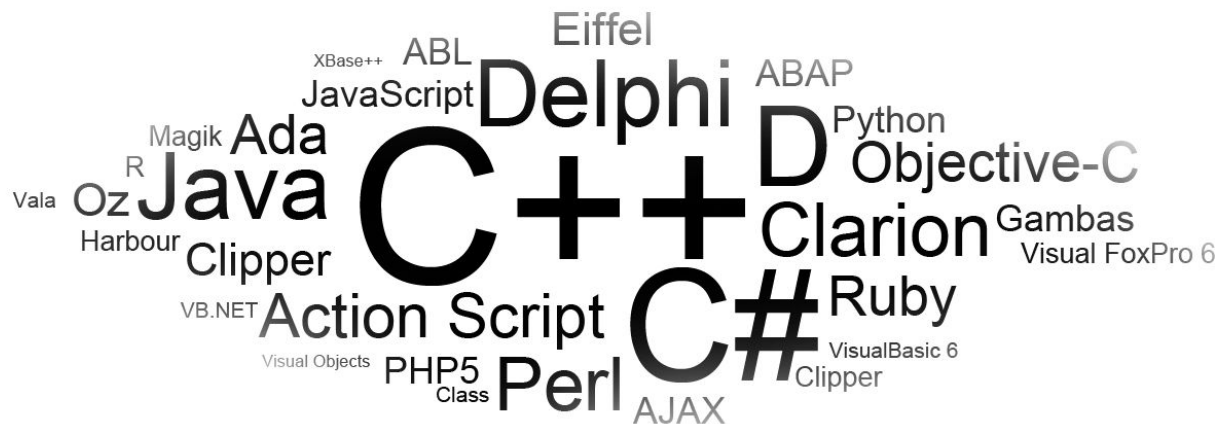
    then max:=a

    else max:=b;

    writeln (max);

End.

Алгоритм, записанный на понятном компьютеру языке программирования, называется **программой**.



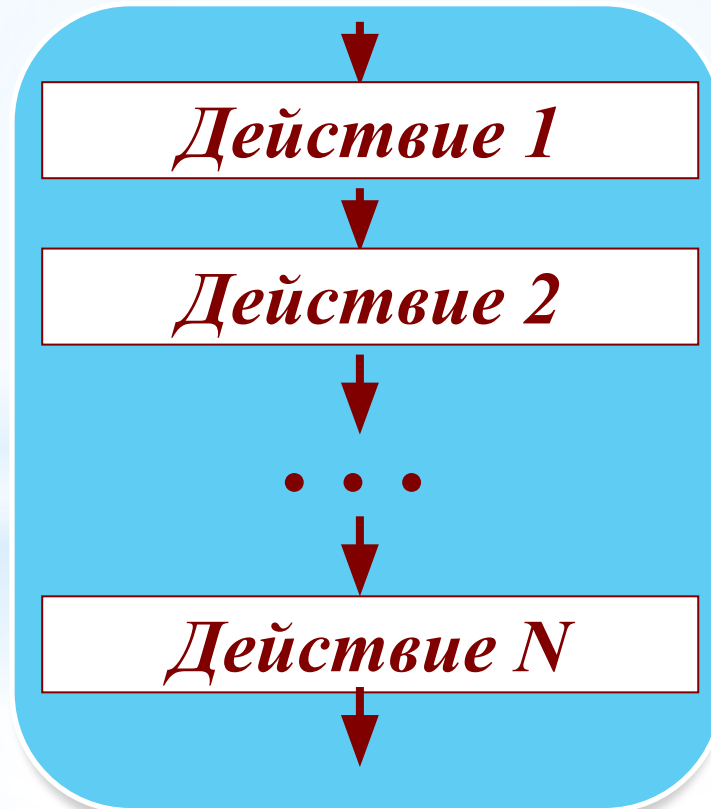
# \* Виды алгоритмов:

- \* Линейный алгоритм - это описание последовательности действий, которые выполняются однократно в заданном порядке.
- \* Разветвляющийся алгоритм - это алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.
- \* Циклический - это описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие (параметр цикла).

# \* Лине́йные алгоритмы

- \* Состоят из нескольких команд (операторов), которые должны быть выполнены последовательно одна за другой

## Структура линейного алгоритма



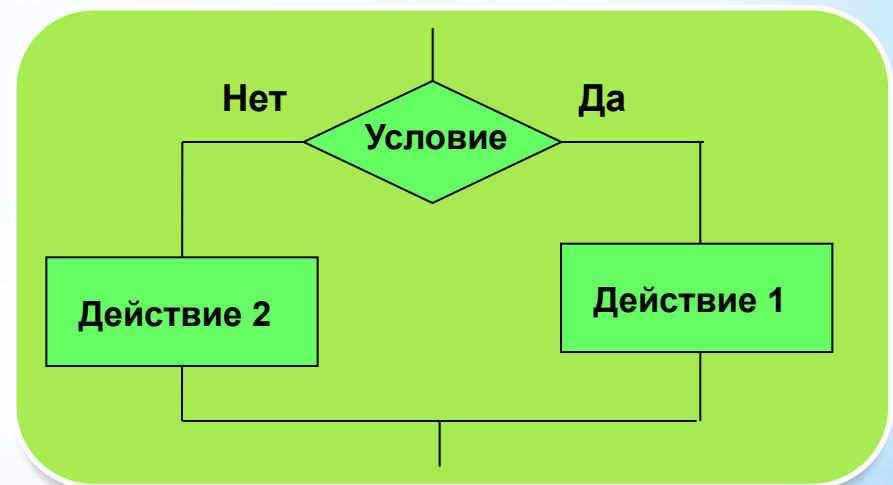


# \* Разветвляющиеся алгоритмы

- \* Состоят из нескольких команд. В зависимости от выполнения некоторого условия совершается одна или другая последовательность шагов.

*Логику принятия решения можно описать так:*

```
ЕСЛИ <условие>  
  ТО <действия 1>  
  ИНАЧЕ <действия 2>  
ВСЕ
```



*Примеры:*

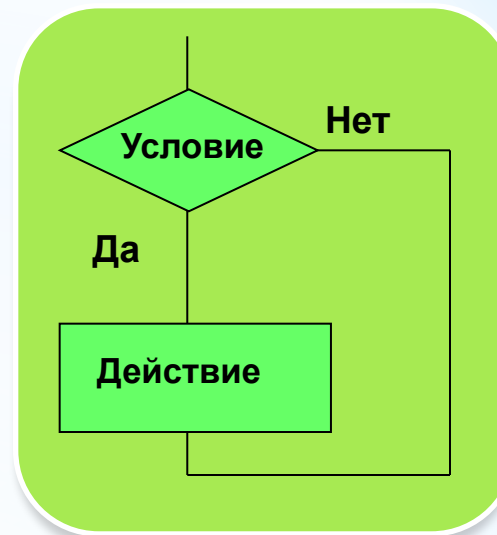
ЕСЛИ хочешь быть здоров, ТО закаляйся, ИНАЧЕ валяйся весь день на диване;

ЕСЛИ низко ласточки летают, ТО будет дождь, ИНАЧЕ дождя не будет;

ЕСЛИ уроки выучены, ТО иди гулять, ИНАЧЕ учи уроки.

**В некоторых случаях <действия 2>  
могут отсутствовать:**

**ЕСЛИ <условие>  
    ТО <действия 1>  
ВСЕ**



**АЛГ Пословица**

**НАЧ**

**ЕСЛИ Назвался груздем**

**ТО Полезай в кузов**

**ВСЕ**

**КОН**

**АЛГ Раскрасить листок**

**НАЧ**

**ЕСЛИ Ты любишь осень?**

**ТО Возьми желтый карандаш**

**ИНАЧЕ Возьми зеленый карандаш**

**ВСЕ**

**Раскрась листок**

**Убери карандаш**

**КОН**

# \* Фальшивая монета



**Задача:** Из трёх монет одинакового достоинства одна фальшивая (более лёгкая). Как её найти с помощью одного взвешивания на чашечных весах без гирь?



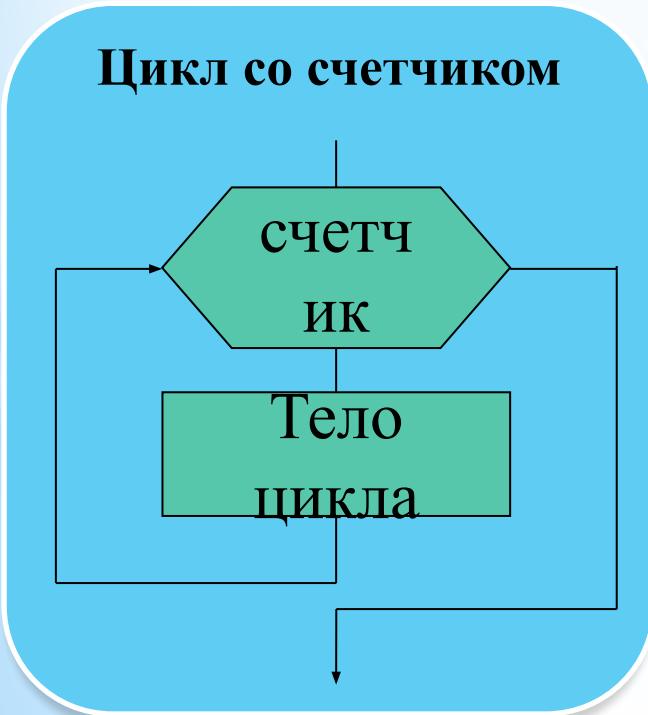
# \* Блок-схема



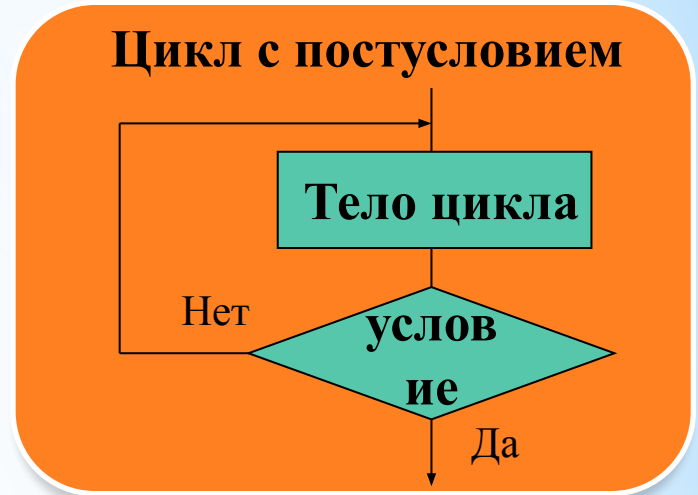
# \* Циклические алгоритмы

- \* Состоят из нескольких команд. Команды повторяются несколько раз (или ни разу) до тех пор, пока выполняется некоторое условие.

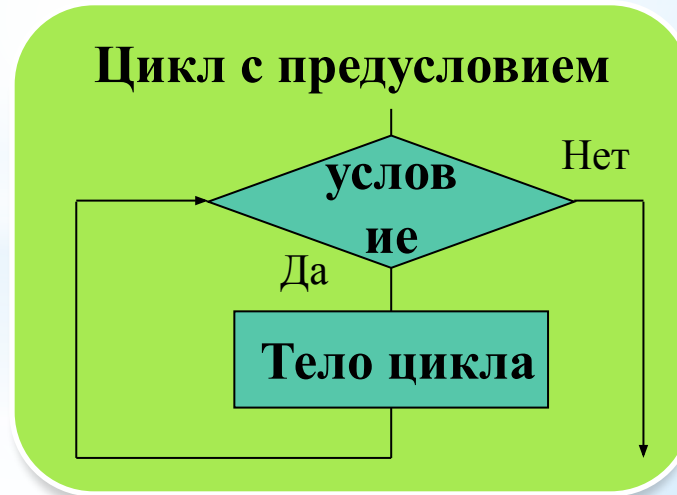
Цикл со счетчиком



Цикл с постусловием



Цикл с предусловием



# \* Подготовка домашнего задания





# Алгоритм поиска Золушки



# \* Самостоятельная работа

1. Семакин И.Г. Базовый уровень: 10 класс. стр. 86-98.
2. Программный принцип работы компьютера.

## Рефераты:

1. Алгоритмы в математике.
2. Мухаммед ибн Муса ал-Хорезми.
3. Алан Тьюринг - создатель теории «логических вычисляющих машин».
4. Андрей Марков и общая теория алгоритмов

## Творческая работа:

1. Приготовление моего любимого блюда.
2. Мой обычный день.