

# АЛГОРИТМИЗАЦІЯ



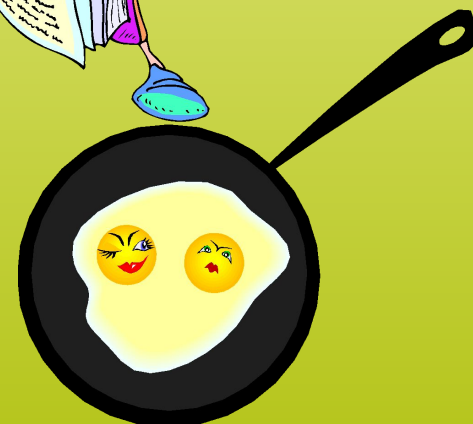
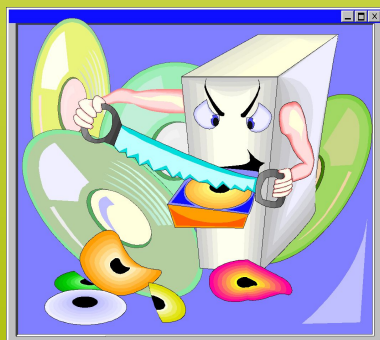
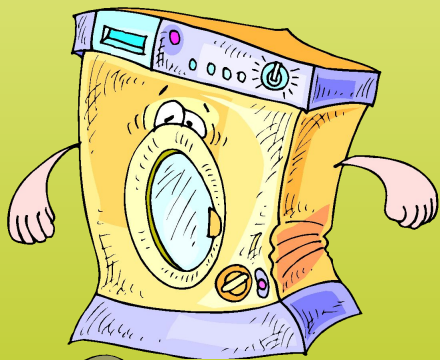
**"Алгоритм"**

# Понятие алгоритма

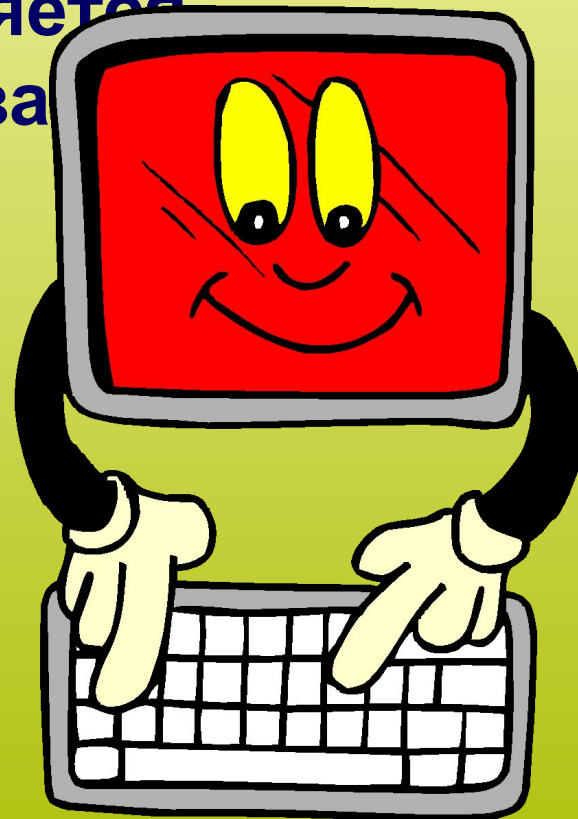
- Слово «алгоритм» происходит от латинского написания имени арабского математика аль-Хорезми (*Algorithmi*), впервые описавший правила выполнения четырёх арифметических



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



**Исполнитель алгоритма** – человек или устройство (в частности, процессор ЭВМ), умеющий выполнять определённый набор действий. Исполнитель является средством реализации алгоритма.



# Исполнитель

Формальный

Неформальный



# Способы записи алгоритмов

- Словесно-формульный (на естественном языке с использованием математических формул)
- Графический (блок-схема)
- На языке программирования (программа)

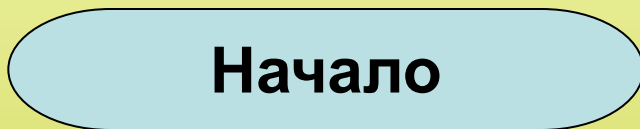
**Примеры.**

СИ, Паскаль, Бейсик и др.

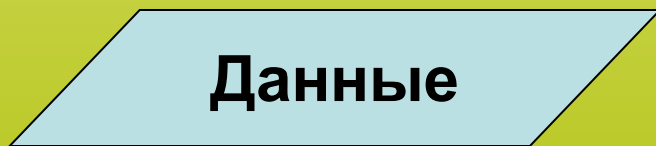


*Блок-схема – это  
графическое  
представление  
алгоритма*

# Блок-схемы



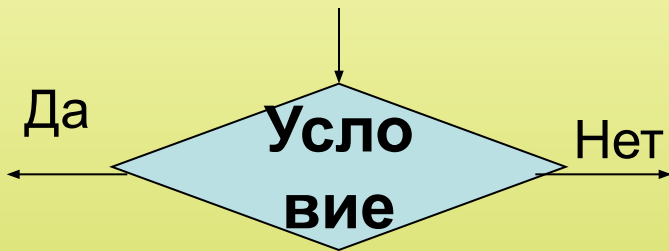
Прямоугольник с закругленными углами, применяется для обозначения начала или конца алгоритма



Параллелограмм, предназначен для описания ввода или вывода данных, имеет один вход вверху и один выход внизу



# Блок-схемы



Ромб, служит для обозначения условий в алгоритмических структурах «ветвление» и «выбор», имеет один вход сверху и два выхода (налево, если условие выполняется, и направо, если условие не выполняется)

**Последовательность**

Прямоугольник, применяется для описания линейной последовательности команд, имеет один вход сверху и один выход внизу

# Свойства алгоритмов

- **Дискретность** — алгоритм должен представлять процесс решения задачи как последовательное выполнение простых шагов.
- **Определенность** — каждое правило алгоритма должно быть четким и однозначным.
- **Понятность** - исполнитель алгоритма должен знать, как его выполнять.
- **Результативность** - алгоритм должен приводить к решению задачи за конечное число шагов.
- **Массовость** – алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными.

**Алгоритмы  
разделяют на:**

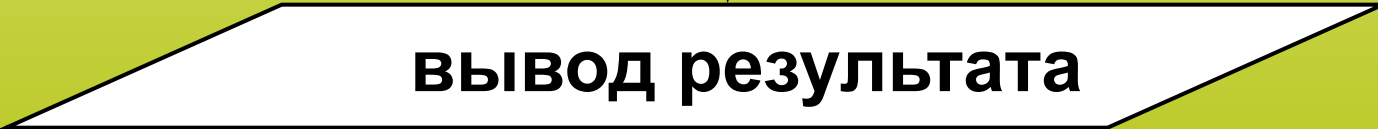
# Линейные алгоритмы

# Линейные алгоритмы

Алгоритм называется **линейным**, если все его действия выполняются последовательно друг за другом от начала и до конца.



■ ■ ■



С  
Е  
Р  
И  
Я

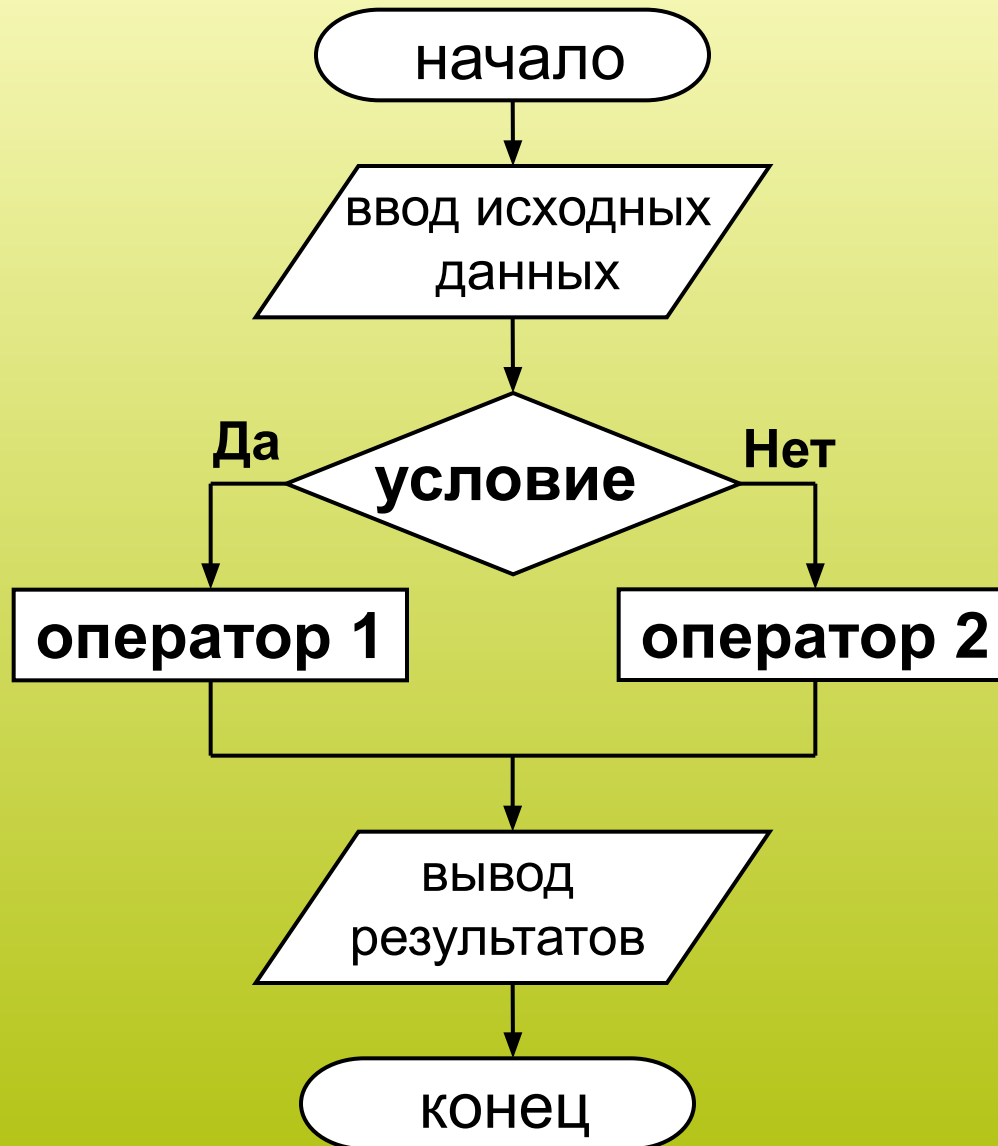
# Пример решения задач



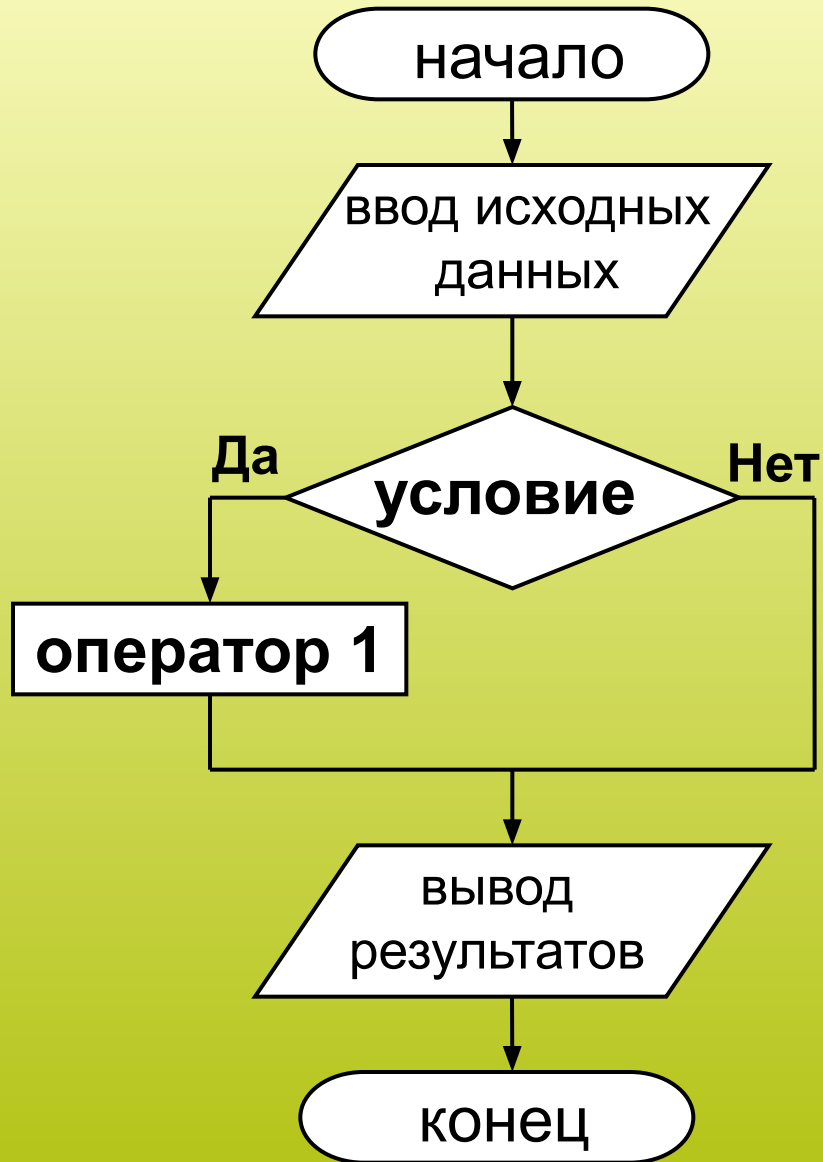
# Разветвляющиеся алгоритмы .

Алгоритм называется  
**разветвляющимся**,  
если порядок выполнения  
шагов алгоритма изменяется в  
зависимости от заданных  
условий.

# Полная команда ветвления



# Неполная команда ветвления



# Примеры решения задач

# Циклические алгоритмы



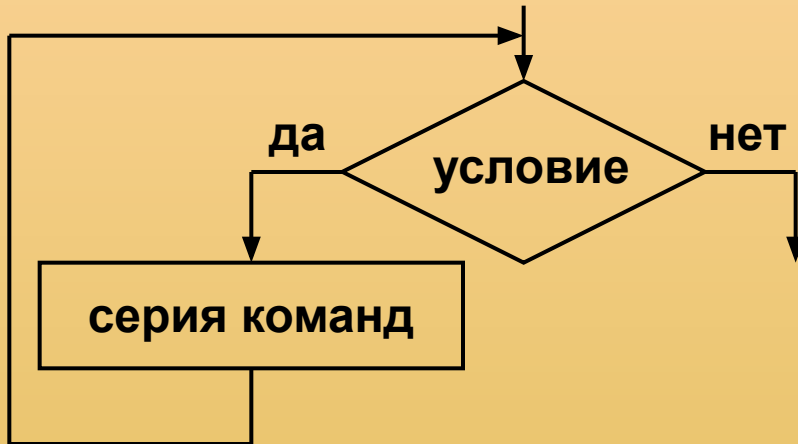
Алгоритм называется

**ЦИКЛИЧЕСКИМ,**

если последовательность шагов  
алгоритма выполняется  
многократно.

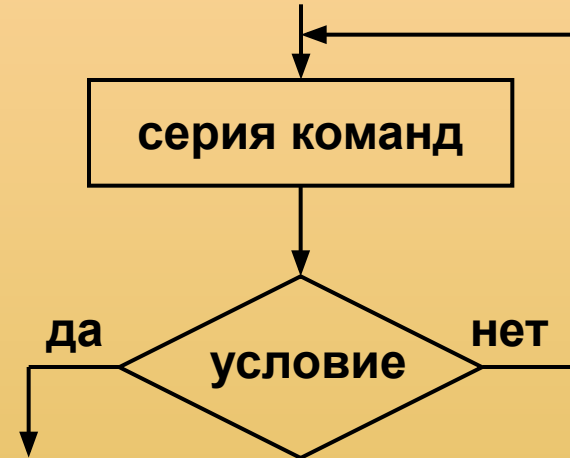
## ЦИКЛ С ПРЕДУСЛОВИЕМ

(цикл-пока)



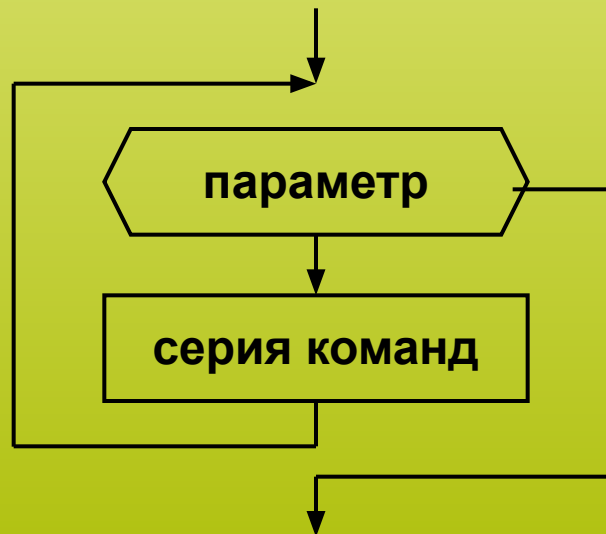
## ЦИКЛ С ПОСТУСЛОВИЕМ

(цикл-до)



*итерационные циклы*

## ЦИКЛ С ПАРАМЕТРОМ





# Примеры решения задач