

# Алгоритмы



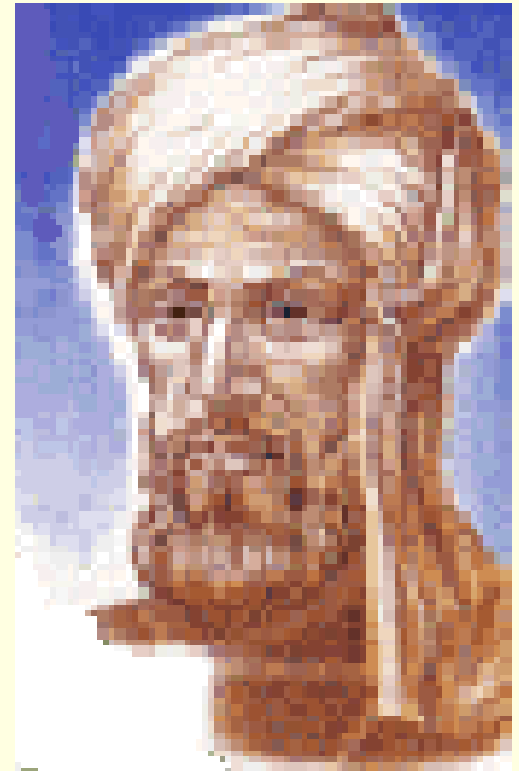
# *Понятие алгоритма*

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

9 век н.э.



***Алгоритм** – это точное и понятное предписание исполнителю совершить последовательность действий над заданными объектами, приводящее исполнителя после конечного числа шагов к достижению указанной цели или решению поставленной задачи.*



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

*Исполнитель является средством реализации алгоритма.*



*Исполнитель*

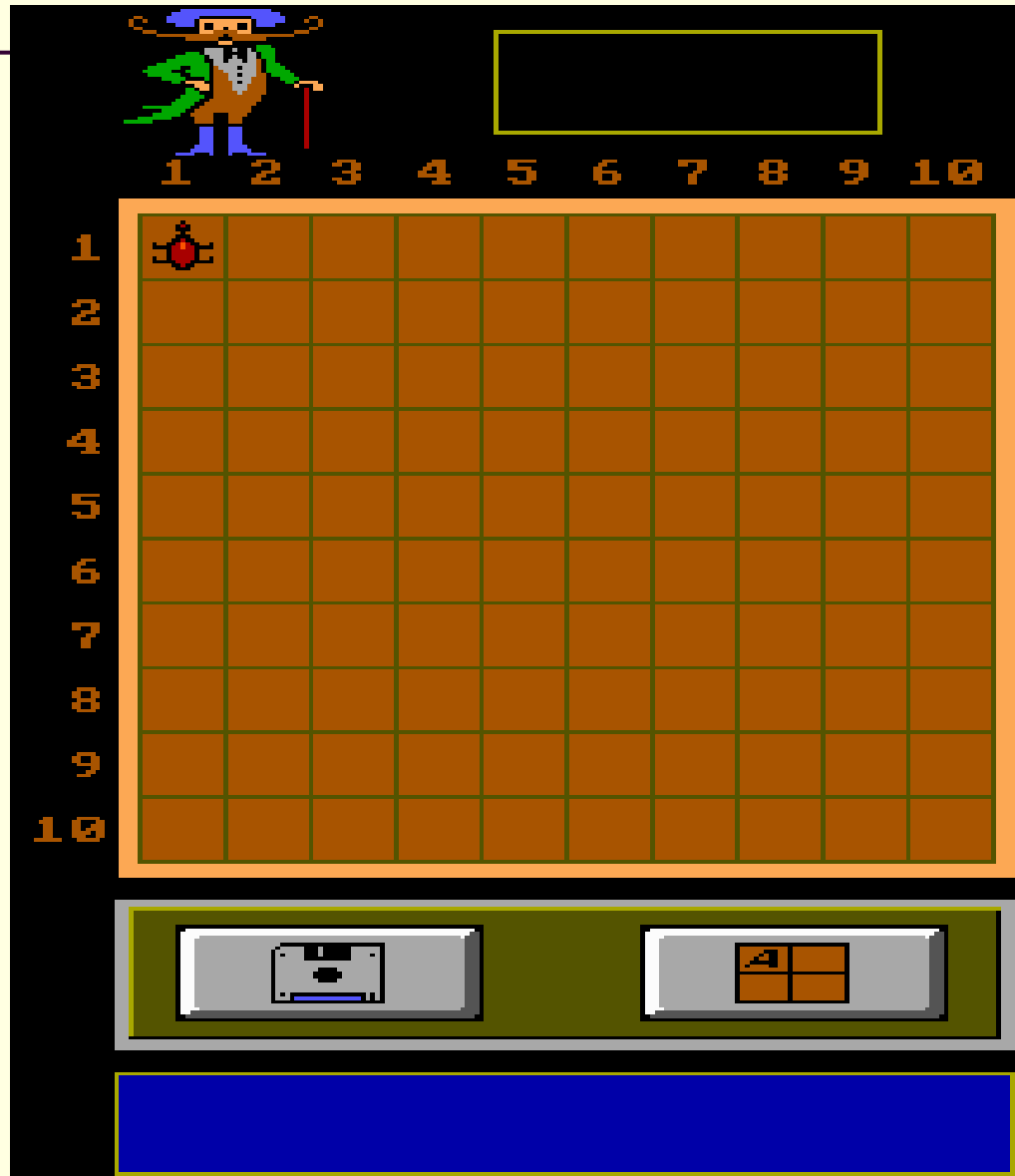
*Формальный*

*Неформальный*



# Исполнителя характеризуют:

- *Среда – это обстановка, в которой работает исполнитель.*



# Исполнителя характеризуют:

- Система команд исполнителя – набор понятных исполнителю команд.



# *Исполнителя характеризуют:*

---

## ***Элементарное действие***

***После вызова команды исполнитель совершает элементарное действие***

## ***Отказы***

***Возникают при вызове команды В недопустимом для данной команды состоянии среды.***



## *Свойства алгоритма:*

---

- 1) дискретность (прерывность)*
- 2) определённости (детерминированность)*
- 3) массовость*
- 4) результативность*
- 5) конечность*
- 6) правильность*

# Критерии качества алгоритма

- **Связанность** – определяется количеством промежуточных результатов, подлежащих запоминанию.
- **Объем алгоритма** – количество операций (шагов), которые необходимо выполнить для достижения конечного результата.
- **Длительность решения** – определяется как количеством, так и сложностью шагов.
- **Разветвленность алгоритма** – характеризует логическую сложность и определяется количеством путей, по которым может реализовываться алгоритм.
- **Цикличность алгоритма** – заключается в том, что фактическое количество операций, которые должны быть выполнены, превышает количество операций, содержащихся в записи алгоритма.

# Способы записи алгоритмов

- *Словесно-формульный (естественный язык) – используется на начальных этапах изучения алгоритмов и предназначен для исполнения алгоритма человеком. Форма записи команд – произвольная.*

## *Пример.*

- *алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел (алгоритм Эвклида).*
- *Алгоритм может быть следующим:*
- *задать два числа;*
- *если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;*
- *определить большее из чисел;*
- *заменить большее из чисел разностью большего и меньшего из чисел;*
- *повторить алгоритм с шага 2.*

*Словесный способ не имеет широкого распространения, так как такие описания:*

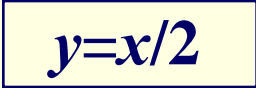
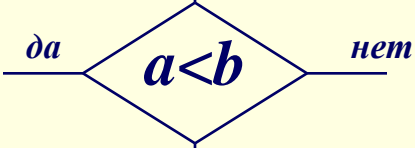
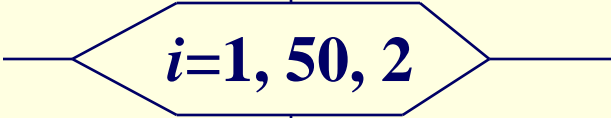
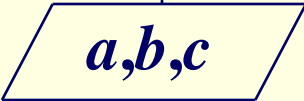


- *строго не формализуемы;*
- *страдают многословностью записей;*
- *допускают неоднозначность толкования отдельных предписаний.*

# Способы записи алгоритмов

---

- *Графический* – это способ представления алгоритма с помощью геометрических фигур (блок – схема).
- 1956 г. – А.А. Ляпунов, Ю.Н. Янов – первое понятие о языке блок – схем алгоритмов.
- ГОСТ 19.002-80

## *Блочные символы (блоки).*

<i>Название блока</i>	<i>Вид блока и пример заполнения</i>	<i>Что обозначает</i>
<b>Процесс</b>		<i>Вычислительное действие</i>
<b>Решение</b>		<i>Проверка условий</i>
<b>Модификация</b>		<i>Начало цикла</i>
<b>Ввод/вывод</b>		<i>Ввод/вывод в общем виде</i>
<b>Пуск/останов</b>		<i>Начало, конец алгоритма</i>
<b>Документ</b>		<i>Вывод результатов на печать</i>

# Алгоритмический язык

*Псевдокод* -представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов.

*Пример.*

- школьный алгоритмический язык в русской нотации (школьный АЯ), описанный в учебнике А.Г. Кушниренко и др. "Основы информатики и вычислительной техники", 1991. Этот язык в дальнейшем мы будем называть просто "алгоритмический язык".

# Алгоритмический язык

## Основные служебные слова

- *алг* (алгоритм)    *сим* (символьный)    дано для да
- *арг* (аргумент)    *лит* (литерный)    надо от нет
- *рез* (результат)    *лог* (логический)    если до при
- *нач* (начало)    *таб*(таблица)    то знач    выбор
- *кон* (конец)    *нц* (начало цикла)    иначе и ввод
- *цел* (целый)    *кц* (конец цикла)    все или вывод
- *вещ* (вещественный)    *длин* (длина)    пока не утв

# Алгоритмический язык

*Общий вид алгоритма:*

*алг* название алгоритма (аргументы и результаты)

*дано* условия применимости алгоритма

*надо* цель выполнения алгоритма

*нач* описание промежуточных величин

| последовательность команд (тело алгоритма)

*кон*



# Программный способ

Язык для записи алгоритма формализован и называется языком программирования. Запись на этом языке называется программой.

Числа, символы, буквы, над которыми производятся те или иные действия называют **операндами**, а указания, предписания, правила преобразования операндов – **операторами**.

**Примеры.**

СИ, Паскаль, Бейсик и др.

# Табличный способ

Наиболее часто используется в экономических расчетах, при выполнении курсовых и лабораторных работ.

*Пример.*

<i>Фамилия</i>	<i>Зарплата</i>	<i>Премия</i>	<i>Всего</i>
<i>Матроскин</i>	<i>5 000</i>	<i>1 500</i>	<i>6 500</i>
<i>Печкин</i>	<i>4 000</i>	<i>1 000</i>	<i>5 000</i>

# Базовые алгоритмические структуры

- *Основные (базовые) структуры алгоритмов – это ограниченный набор блоков и стандартных способов их соединения для выполнения типичных последовательностей действий.*
- *Структурный подход к разработке алгоритмов предполагает использование только нескольких основных структур, комбинация которых дает все многообразие алгоритмов и программ.*



# Структура СЛЕДОВАНИЕ

*Школьный алгоритмический  
язык*

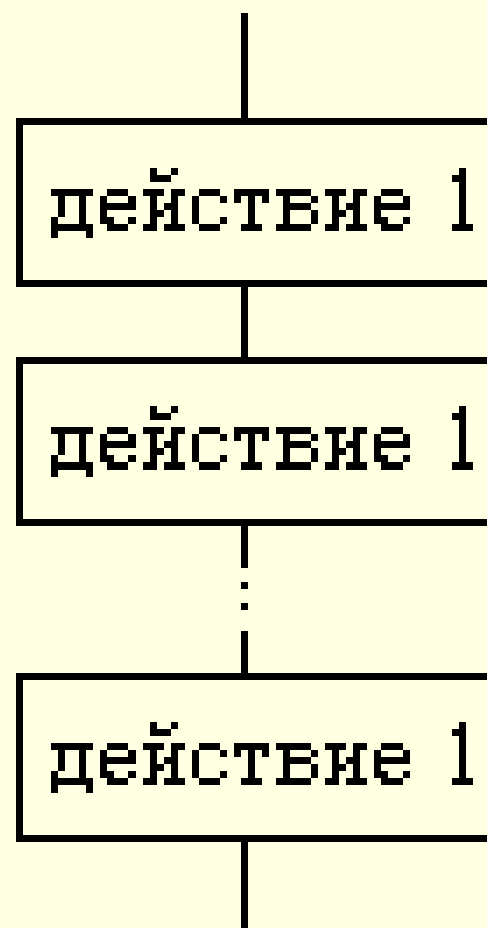
*Действие 1*

*Действие 2*

*.....*

*Действие N*

*Язык блок-схем*



# Структура ВЕТВЛЕНИЕ

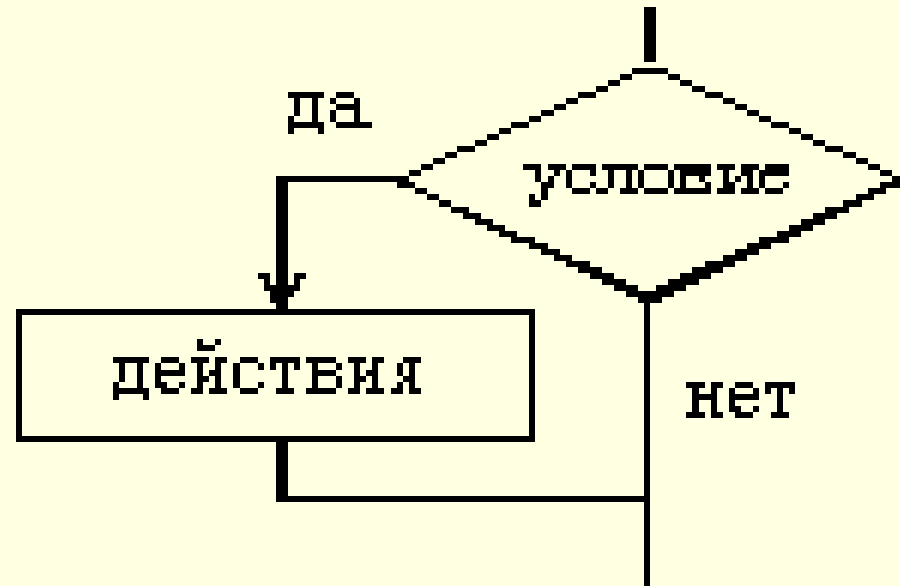
*если – то*

*Если условие*



*то действие*

*Все*



# Структура ВЕТВЛЕНИЕ

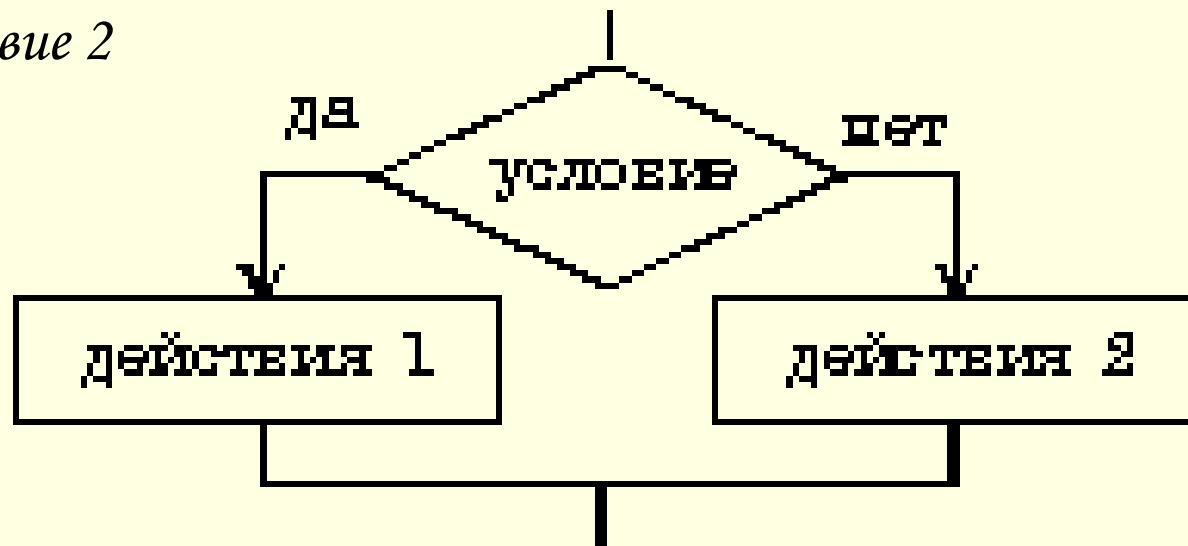
*если – то – иначе*

*Если условие*

*то действие 1*

*иначе действие 2*

*Все*



# Структура ВЕТВЛЕНИЕ выбор - иначе

выбор

при условии 1: действия 1

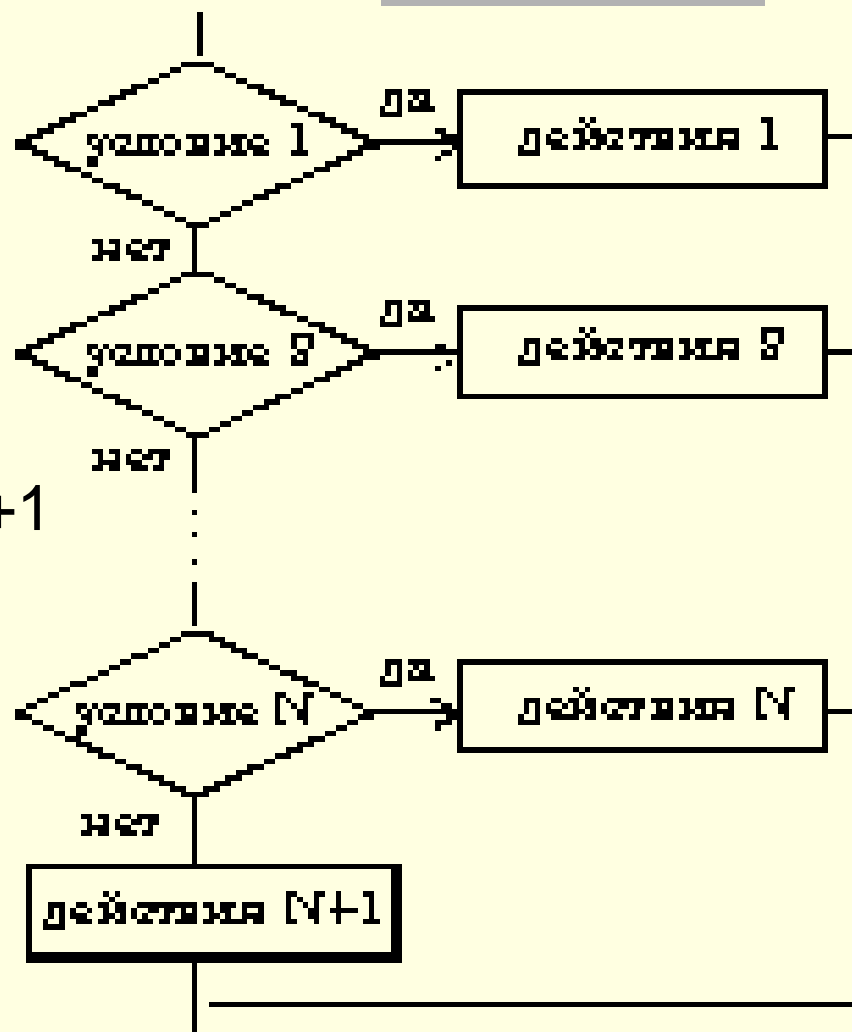
при условии 2: действия 2

.....

при условии N: действия N

иначе

действия N+1



все

# Структура ВЕТВЛЕНИЕ выбор

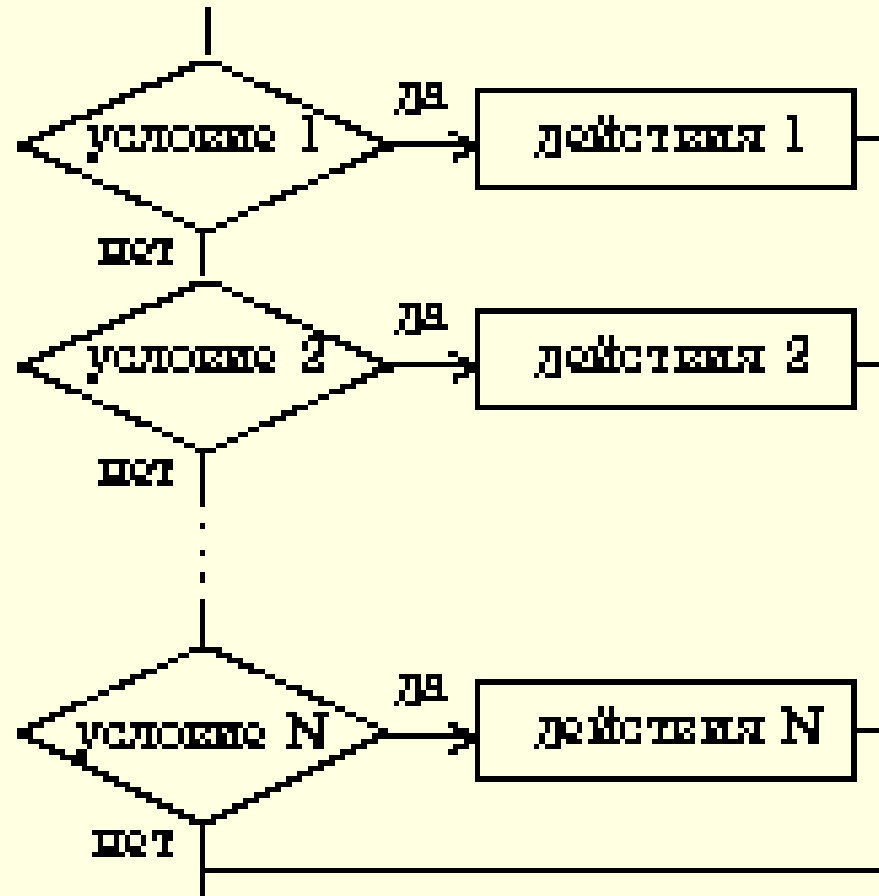
выбор

при условии 1: действия 1

при условии 2: действия 2

.....

при условии N: действия N



все



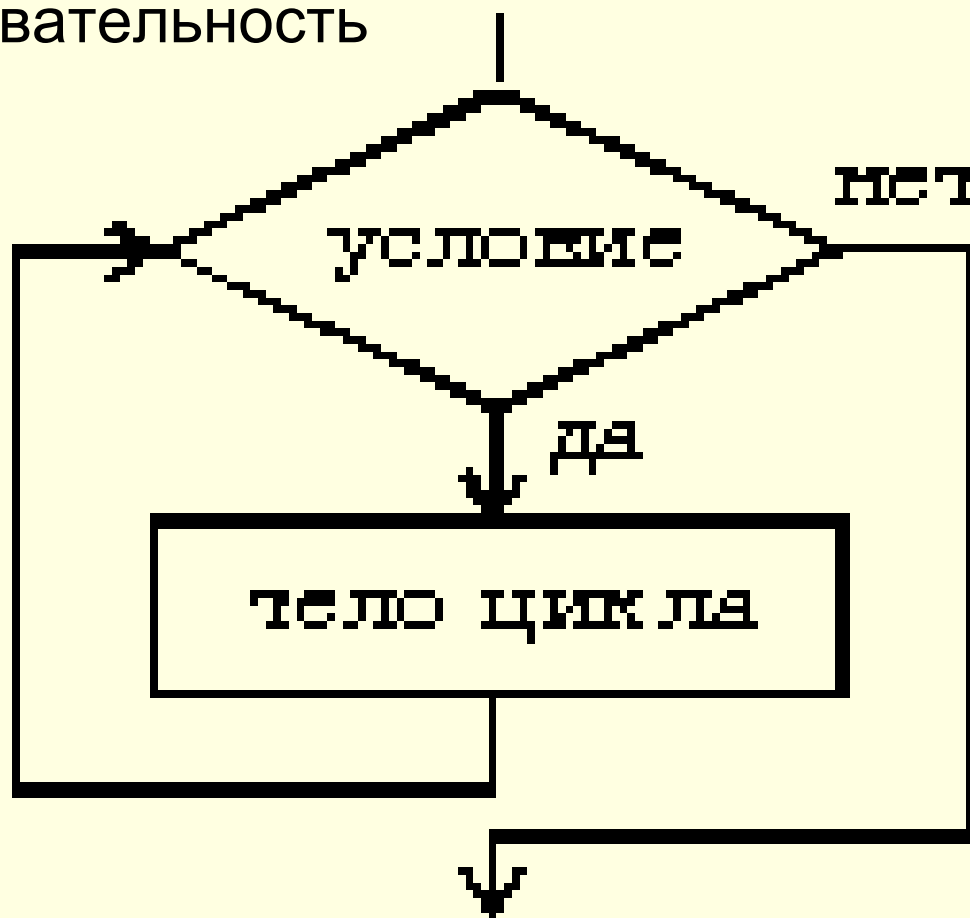
# Структура ЦИКЛА

## Цикл типа ПОКА (с предусловием)

НЦ пока условие

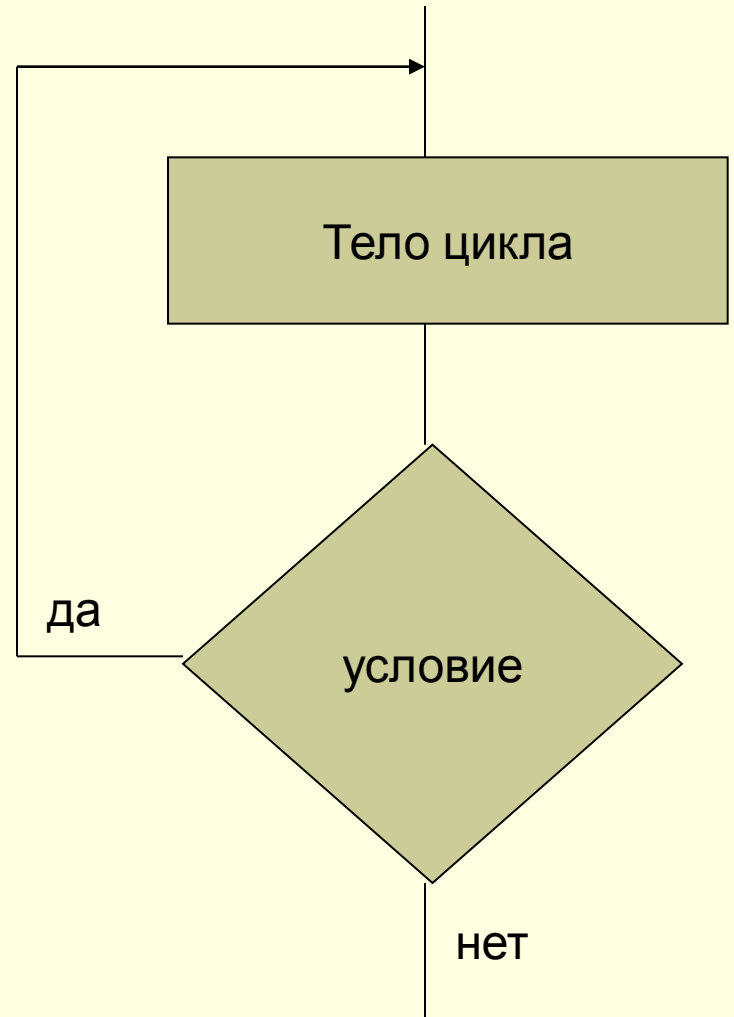
тело цикла (последовательность действий)

КЦ



# Структура ЦМКА

## Цикл типа DO (с постусловием)



# Структура ЦИКЛА

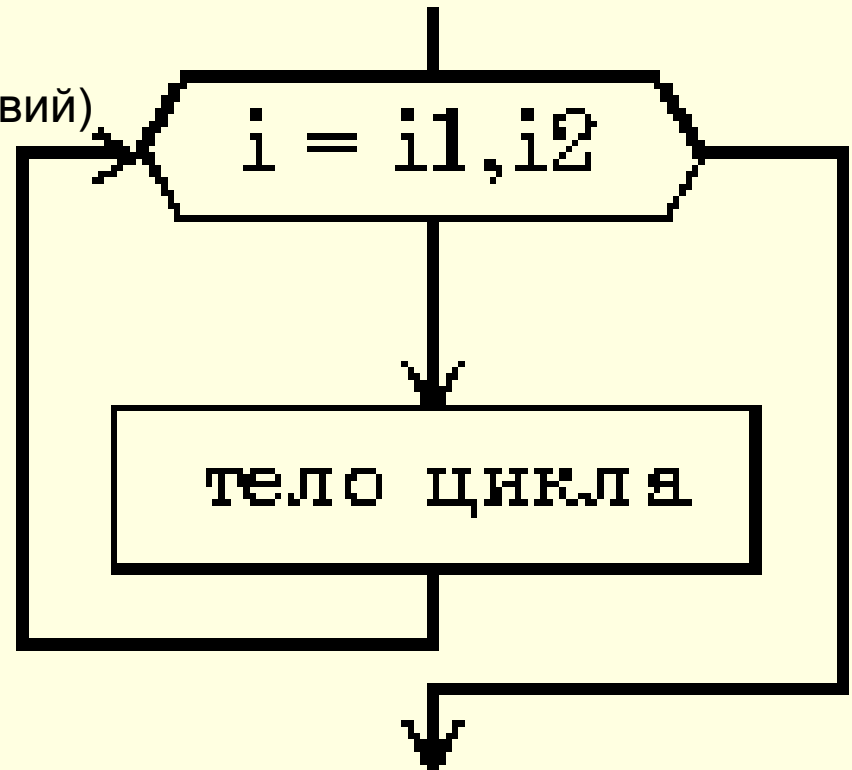
## Цикл типа ДЛЯ (с параметром)

---

нц для  $i$  от  $i_1$  до  $i_2$

тело цикла (последовательность действий)

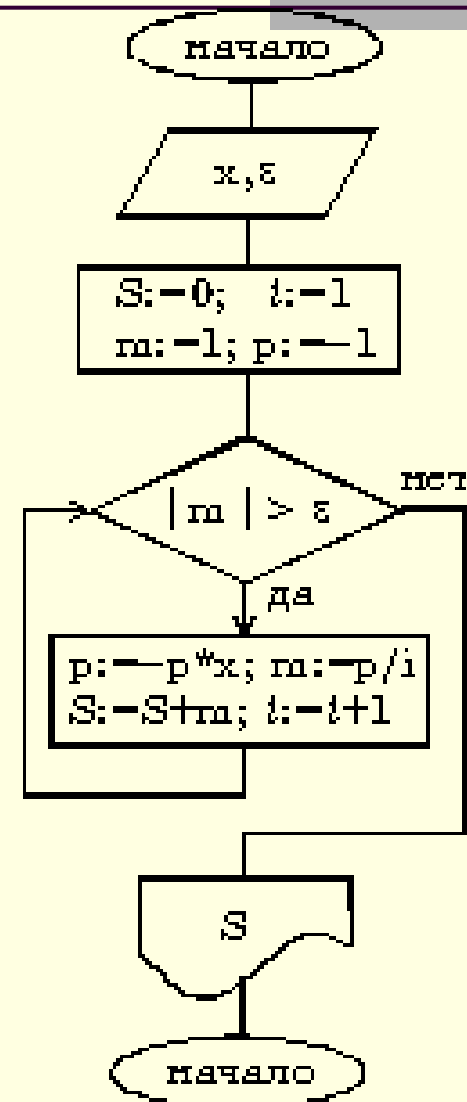
кц



# Итерационный цикл

Особенностью итерационного цикла является то, что число повторений операторов тела цикла заранее неизвестно. Для его организации используется цикл типа **пока**. Выход из итерационного цикла осуществляется в случае выполнения заданного условия.

В итерационных алгоритмах необходимо обеспечить обязательное достижение условия выхода из цикла (сходимость итерационного процесса). В противном случае произойдет "зацикливание" алгоритма, т.е. не будет выполняться основное свойство алгоритма — **результативность**.

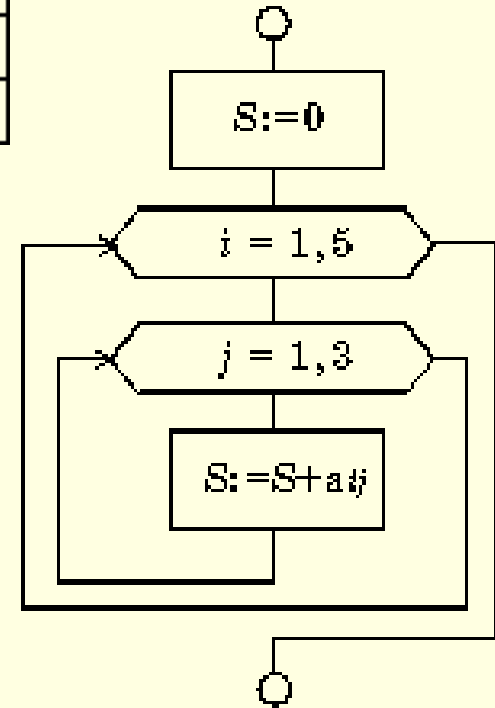


# Вложенный цикл

- Возможны случаи, когда внутри тела цикла необходимо повторять некоторую последовательность операторов, т. е. организовать внутренний цикл. Такая структура получила название цикла в цикле или вложенных циклов. Глубина вложения циклов (то есть количество вложенных друг в друга циклов) может быть различной.

	1	2	3
1			
2			
3			
4			
5			

```
S := 0;  
нц для i от 1 до 5  
  нц для j от 1 до 3  
    S:=S+A[i,j]  
  кц  
кц
```



# Примеры алгоритмических матришек

потом переход к новому повторению по наружной петле.

