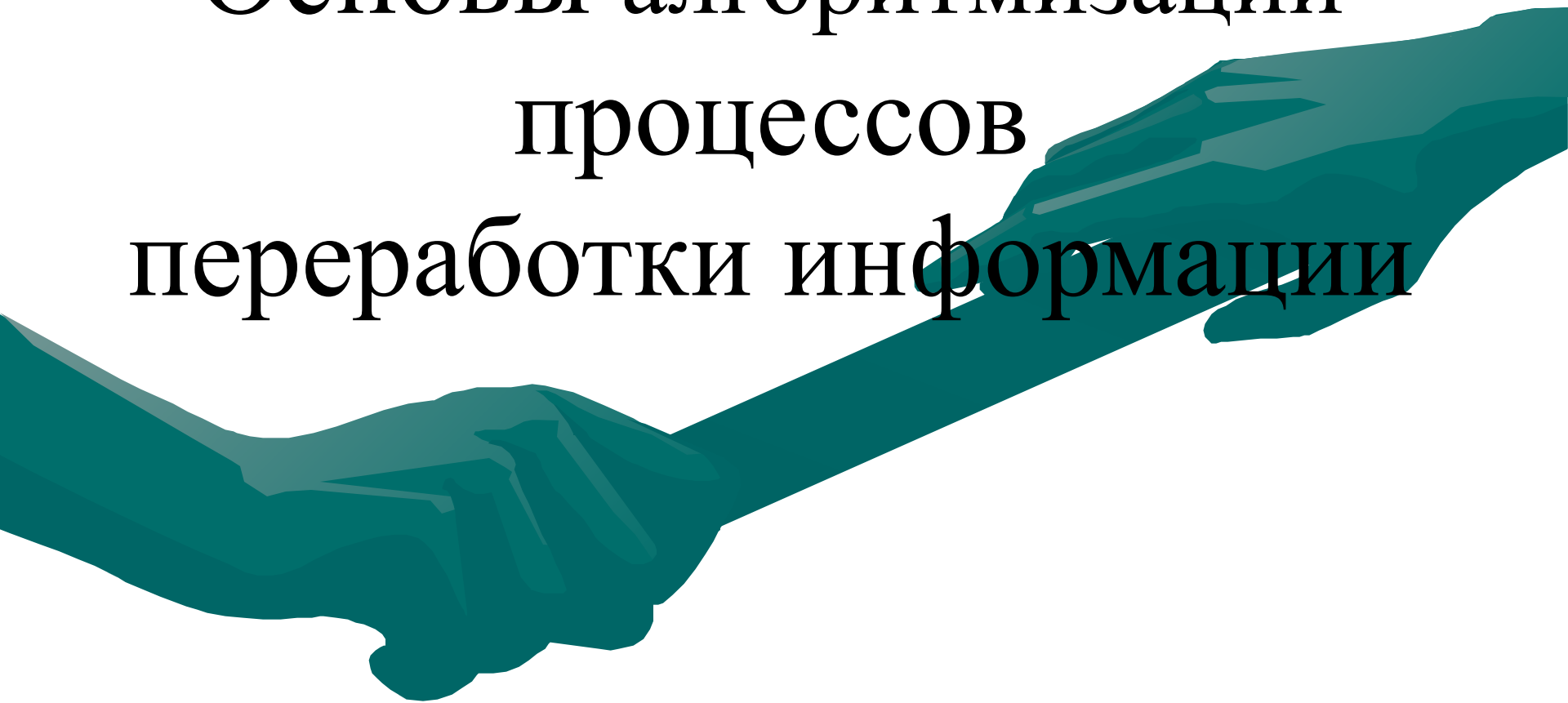


ОСНОВЫ алгоритмизации процессов переработки информации



1. Этапы решения задач на ЭВМ

- 1) Разработка математической модели решаемой задачи;
- 2) Выбор, либо разработка метода решения;
- 3) Разработка алгоритма, запись его на некотором языке (ЕЯ, языке блок-схем и т.п.);
- 4) Программирование решения задачи на одном из языков программирования;
- 5) Тестирование и отладка программы или комплекса программ;
- 6) Решение задачи на ЭВМ.

Решение задач на ЭВМ включает:

- 1) Подготовку исходных данных для программы (осуществляет пользователь);
- 2) Запуск программы (пользователь);
- 3) Производство необходимых действий в соответствии с программой (ЭВМ);
- 4) Выдача полученных результатов (ЭВМ).
- 5) Анализ результатов решения задачи (пользователь).

Программы, написанные на языке Паскаль, перед выполнением на ЭВМ должны транслироваться в машинные программы, понятные компьютеру. Для этого используются специальные программы – компиляторы.

2. Понятие алгоритма

(фундаментальное понятие информатики)

- **Алгоритмом** называется чёткое описание последовательности действий, которое необходимо выполнить для решения задачи.
- **Алгоритм** – это понятное и точное предписание исполнителю совершить определённую последовательность действий для достижения указанной цели или решения поставленной задачи.
- Разработать **алгоритм** решения задачи, значит, разбить задачу на последовательно выполняемые шаги (этапы).

Пример алгоритма на ЕЯ:

- **Исходные данные:** $a=1$, $b=3$, $c=2$.
- **Алгоритм:**
 1. Вычислить $D=b^2 - 4ac$;
 2. Сравнить D с нулём. Если $D < 0$, перейти к п.3.
В противном случае вычислить и напечатать:
 $x_1 = (-b + \sqrt{D}) / (2a)$, $x_2 = (-b - \sqrt{D}) / (2a)$. Перейти к п.4.
 3. Напечатать сообщение “*Уравнение не имеет действительных корней*”.
 4. Прекратить вычисления.
- Выполнив указанную последовательность для заданных значений a , b , c , получим решение квадратного уравнения $x^2 + 3x + 2 = 0$.

3. Свойства алгоритма

1) Дискретность.

Алгоритм должен представлять процесс решения задачи как последовательное выполнение простых шагов (этапов). При этом для выполнения каждого шага требуется некоторый конечный отрезок времени. Таким образом, преобразование исходных данных в результат осуществляется во времени дискретно.

2) Определенность (Детерминированность).

Каждое правило алгоритма должно быть четким, однозначным и не оставлять место для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер, не требует никаких дополнительных указаний или сведений и может быть поручено любому исполнителю.

3) **Результативность** (Конечность).

Свойство состоит в том, что алгоритм должен приводить к решению задачи за конечное число шагов.

4) **Массовость.**

Алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными.

При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

Например, алгоритм решения квадратного уравнения применим для различных наборов коэффициентов **a**, **b**, **c** **a**≠0.

4. Способы записи алгоритма

4.1. Запись на естественном языке

1) Этап обработки (вычисления):

$V := \text{Выражение}$,

где V – переменная; **выражение** – задает правила вычисления значения, которое далее будет присвоено переменной V (это может быть, например, знакомое нам алгебраическое (арифметическое) выражение).

2) Проверка условия:

Если <условие> идти к N .

Если условие удовлетворяется, выполняется переход к этапу N , иначе переход к следующему по порядку этапу.


3) Переход к этапу с номером N : *Идти к N .*

4) Конец вычислений: *Закончить вычисления.*

4.2. Изображение алгоритма в виде блок-схемы

- ❖ Блок-схемой называется наглядное графическое изображение алгоритма, когда отдельные этапы (действия алгоритма) изображаются при помощи различных геометрических фигур (**блоков**), а связи между этапами указываются при помощи стрелок, которые соединяют эти фигуры и определяют последовательность выполнения этапов.
- ❖ Существует ГОСТ, определяющий правило выполнения блок-схем и обозначения для отдельных операций (этапов) процесса обработки данных (ГОСТ 19.002-80 и ГОСТ 19.003-80).

Обозначения основных операций :



**Содержание
этапа**

Этап обработки (вычисления).

Внутри прямоугольника записывается содержание этапа.

Проверка условия.

Внутри ромба записывается условие.

Если условие выполняется, то следующим выполняется этап по стрелке «**Да**», иначе по стрелке «**Нет**». В результате осуществляется выбор одного из двух возможных путей реализации вычислительного процесса.



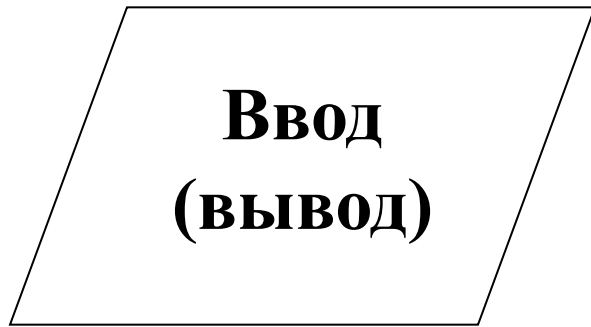
**Начало,
конец**

Начало, конец процесса обработки информации. Внутри записывается «начало» или «конец».

**Имя
подпрограммы
(параметры,
при
которых она
должна быть
выполнена)**

**Ранее созданные и описанные
отдельно алгоритмы и
программы (подпрограммы).**

Внутри указывается имя подпрограммы и параметры, при которых она должна быть выполнена.



Ввод исходных данных и вывод результатов.

Внутри параллелограмма пишется слово «**Ввод**» или «**Вывод**», и перечисляются переменные, подлежащие вводу или выводу.

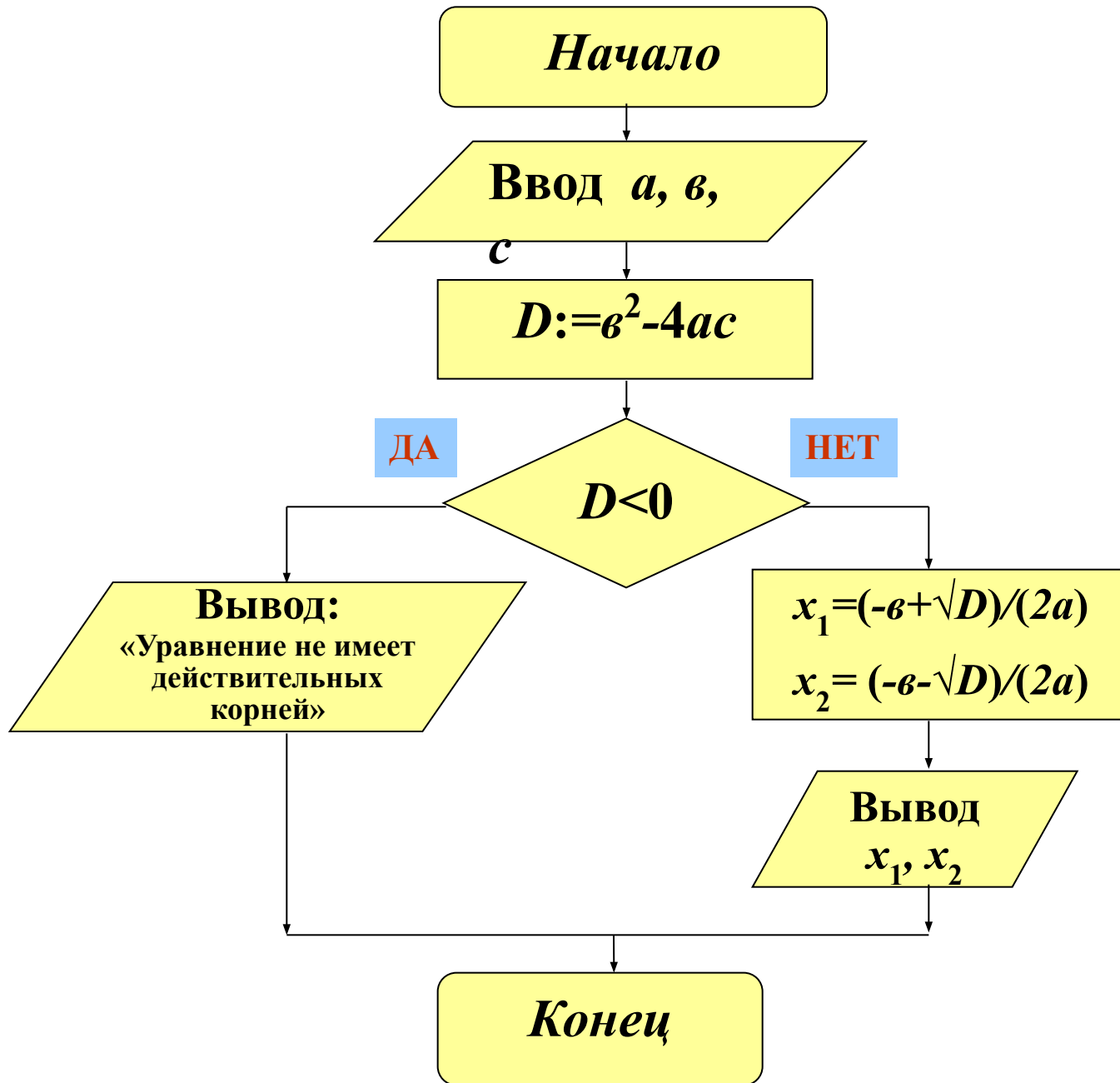


Модификатор.

Внутри шестигранника записывается алгоритм изменения параметра.

Используется, в частности, для обозначения оператора цикла с параметром.

Пример блок-схемы алгоритма



4.3. Понятие об алгоритмических языках

- Алгоритмический язык (АЯ) – это система обозначений и правил для единообразной и точной записи алгоритмов и их исполнения.
- АЯ близки к естественному языку. Однако, правило построения конструкций в АЯ более «жесткие».
- Это означает, что АЯ допускают меньшее разнообразие для описаний действий алгоритма, чем ЕЯ и привычная математическая символика, поэтому компьютер однозначно понимает любую конструкцию языка. Например, для умножения двух переменных ***a*** и ***b*** общепринятая математическая символика допускает несколько возможных форм записи: ***ab***; ***axb***; ***a · b***.
- На АЯ, например на Паскале, эту операцию можно записать только единственным образом: ***a * b***.

5. Основные структуры алгоритмов

- **Основные структуры алгоритмов** — это *ограниченный набор блоков и стандартных способов их соединения для выполнения типичных действий.*

Рассмотрим основные структуры (схемы), которые рекомендуются при использовании структурного подхода к разработке алгоритмов и программ.

5.1. Линейная структура.

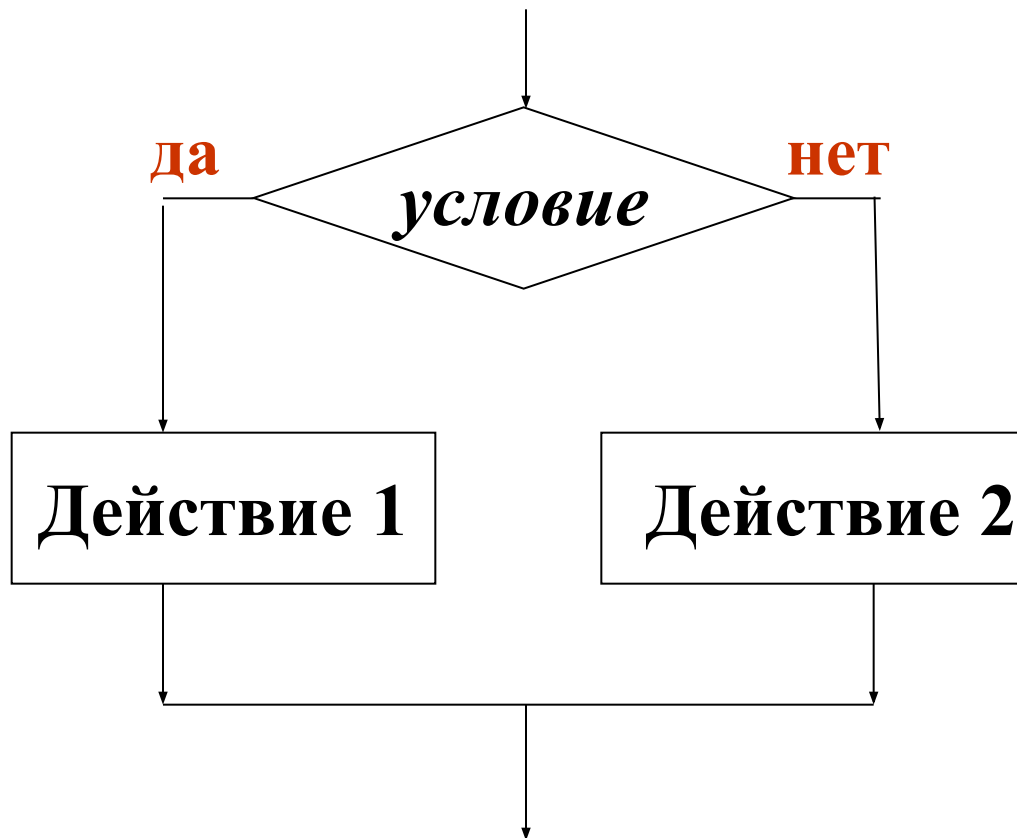
Представляет собой последовательное размещение блоков и групп блоков. В программе реализуется последовательным размещением операторов.

Примечание: В языках программирования предписание о выполнении некоторого **действия** (**операции**) называется **оператором**.

5.2. Разветвленная структура.

Применяется, когда в зависимости от условия нужно выполнить либо одно, либо другое действие.

Соответствует схеме:



Оператор АЯ

Паскаль:

IF условие **THEN**

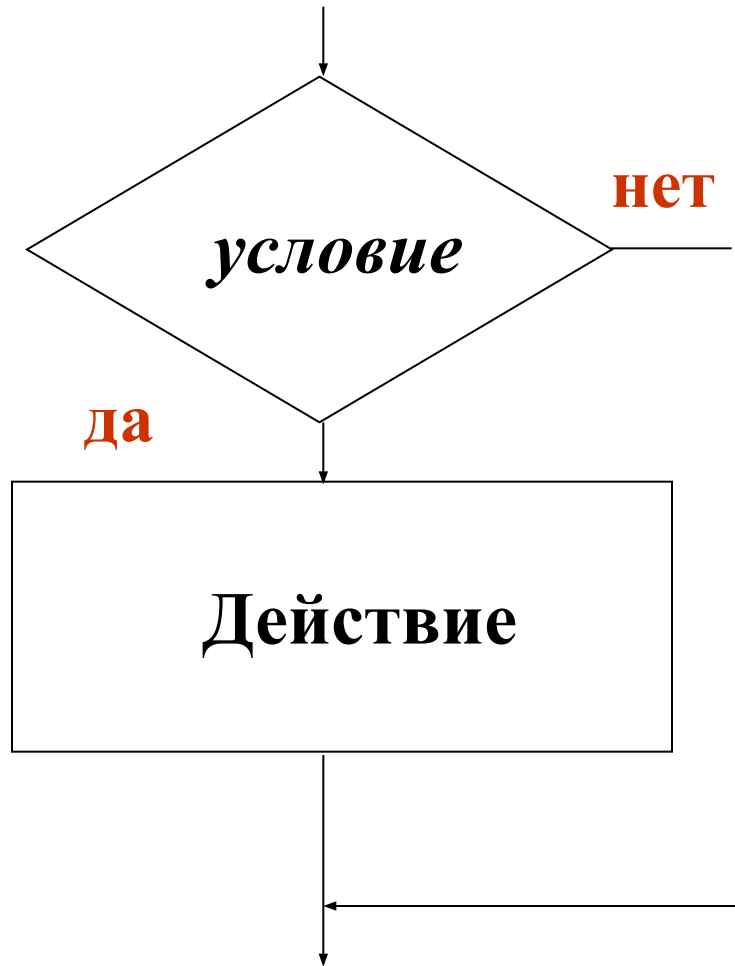
Оператор 1

ELSE *Оператор 2;*

Действия 1, 2 могут, в свою очередь, содержать несколько этапов.

Обход – частный случай разветвления, когда одна ветвь не содержит никаких действий.

Схема имеет вид:



Оператор АЯ
Паскаль:

IF условие **THEN**
Оператор;

5.3. Циклические структуры

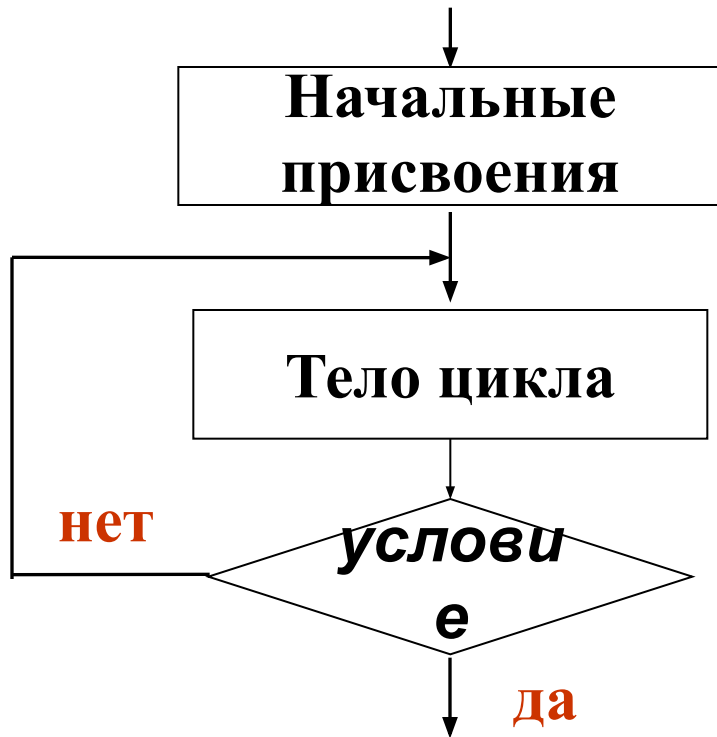
1) Цикл “ДО” (**REPEAT**) – цикл с постусловием.

Тело цикла – это последовательность действий (операторов), которая выполняется многократно.

Начальные присвоения – это задание начальных значений тем переменным, которые используются в цикле.

Применяется при необходимости выполнить какие-либо операторы несколько раз **до выполнения некоторого условия**.

Отличается тем, что он всегда выполняется хотя бы один раз, так как проверка условия выхода из цикла происходит **после** того, как тело цикла выполнено.



Оператор АЯ

Паскаль:

REPEAT

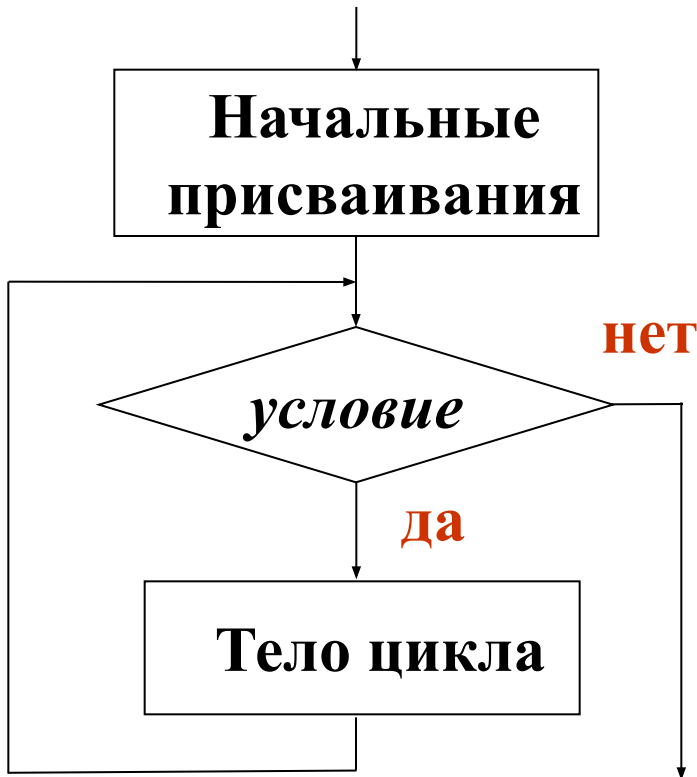
Операторы тела цикла

UNTIL условие;

2) Цикл “ПОКА” (**WHILE**) – цикл с предусловием.

Применяется при необходимости выполнить какие-либо вычисления (операторы) несколько раз **до тех пор пока выполняется некоторое условие**.

Особенность цикла в том, что проверка условия проводится **до** выполнения операторов тела цикла, и, если при первой проверке условие выхода из цикла выполняется, то **операторы тела цикла не выполняются ни разу**.



Оператор АЯ

Паскаль:

WHILE *условие* **DO**

BEGIN

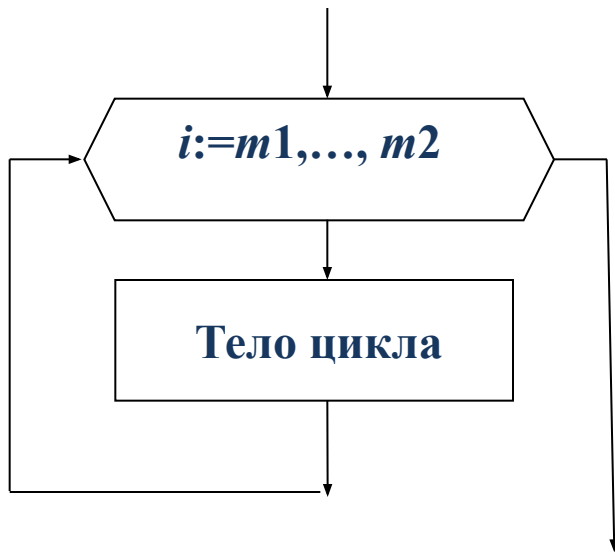
Операторы тела цикла

END;

3) Цикл “ДЛЯ” (FOR) – цикл с параметром.

Используется, когда **заранее известно, сколько раз должны повториться операторы тела цикла.**

Циклу “FOR” соответствует следующая схема алгоритма:



Операторы АЯ Паскаль:

а) FOR $i:=m1$ TO $m2$ DO
BEGIN

Операторы тела цикла

END; { $m2 > m1$ }

б) FOR $i:=m1$ DOVNTO $m2$ DO
BEGIN

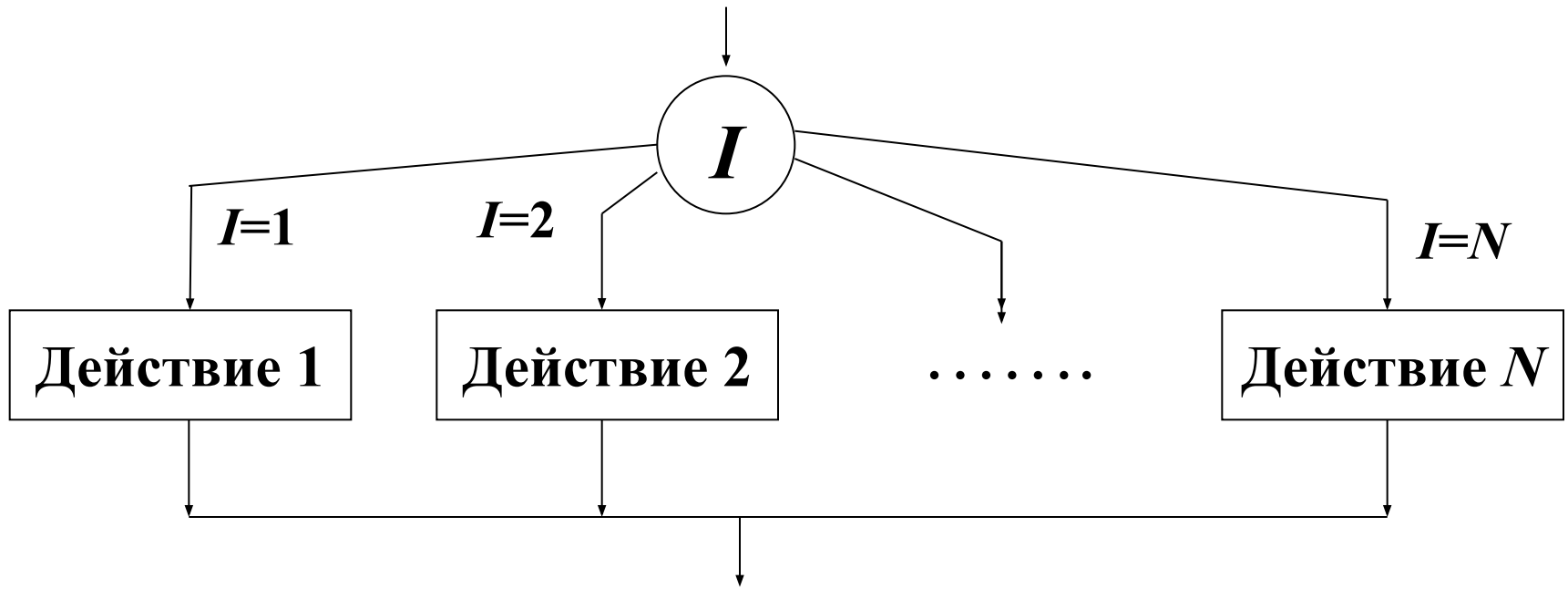
Операторы тела цикла

END; { $m2 < m1$ }

FOR (для), **TO** (до), **DOVNTO** (вниз до), **DO** (выполнить) – служебные слова; i – параметр цикла; $m1$, $m2$ – начальное и конечное значения параметра цикла.

В качестве параметра цикла i может быть только переменная, а в качестве $m1$ и $m2$ – выражения порядкового типа.

5.4. Структура множественного выбора



Оператор АЯ Паскаль:

CASE <выражение> **OF**

Константа 1: оператор 1;

• • •

Константа N: оператор N;

END [**ELSE** оператор];

В зависимости от значения I выбирается одно из $I \in \overline{1, N}$ действий (одна из групп операторов, помеченных константами $I \in \overline{1, N}$).

Внимание!

Все базовые структуры обладают **общим свойством**: **имеют один вход и один выход**, поэтому могут последовательно соединяться между собой.

Алгоритм решения задачи можно построить путем соединения рассмотренных выше **базовых алгоритмических структур**.

При этом соединяться между собой эти структуры могут двумя способами:

последовательным и вложенным.

(представление о глубине вложенности).