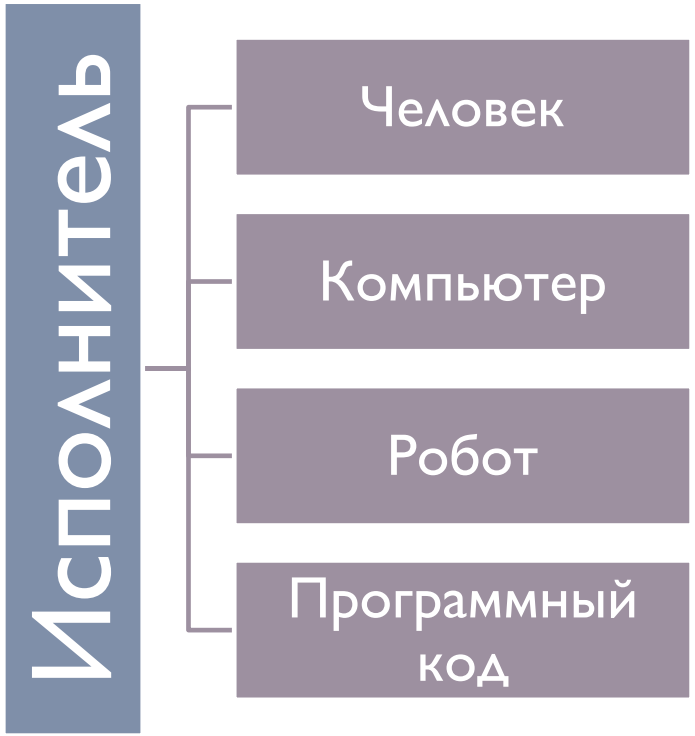
АЛГОРИТМЫ. АЛГОРИТМИЗАЦИЯ.

ПОНЯТИЕ АЛГОРИТМА И ЕГО СВОЙСТВА

Алгоритм (Мухаммеда ибн Муса ал-Хорезми (Alhorithmi), 783—850 гг.) — заранее заданное, понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения решения задачи за конечное число шагов.

Алгоритмы предназначены для выполнения некоторым исполнителем.

Исполнитель алгоритма – абстрактная или реальная система, способная выполнить действия, предписанные алгоритмом.



Исполнителя характеризуют:

- Среда;
- Элементарные действия;
- Система команд;
- Отказы.

СВОЙСТВА АЛГОРИТМА

Формальность (понятность для исполнителя) – исполнитель алгоритма должен понимать, как его выполнять.

Дискретность (прерывность, раздельность) –каждый алгоритм состоит из отдельных законченных действий.

Определенность (детерминированность, точность) –каждый шаг алгоритма должен не допускать различных толкований.

Результативность (конечность) – алгоритм должен завершаться за конечное число шагов.

Массовость – применимость алгоритма ко всем задачам рассматриваемого типа.

ФОРМЫ ПРЕДСТАВЛЕНИЯ АЛГОРИТМОВ

Формы

- Словесная (представляет структуру алгоритма на естественном языке)
- Графическая (изображение из графических символов – блок-схема)
- Псевдокод (описание структуры алгоритма на естественном, частично формализованном языке)
- Программа (описание структуры алгоритма на языке алгоритмического программирования)

СЛОВЕСНОЕ ОПИСАНИЕ АЛГОРИТМА

Словесное описание алгоритма представляет собой запись алгоритма в произвольной форме на естественном, например, русском языке.

Достоинства	Недостатки
• Возможность лучше описать отдельные операции; • Можно описать любой алгоритм.	• Многословность; • Отсутствие наглядности; • Строго не формализуем; • Неоднозначность толкования.

ПРИМЕР: НАЙТИ НАИБОЛЬШЕЕ ЧИСЛО ИЗ ТРЕХ ЗАДАННЫХ (A, B, C) (СЛОВЕСНОЕ ОПИСАНИЕ)

Сравнить a и b.

- Сравнить первое и второе число (a и b). Переменной *max* присвоить значение переменной, содержащей большее значение.

Сравнить max и c.

- Сравнить значение переменной *max* с третьим числом (c). Если значение c окажется больше, чем *max*, то присвоить *max* значение третьего числа. Если же значение *max* окажется больше, то выполняется след. шаг.

max результат.

- Принять max в качестве результата.

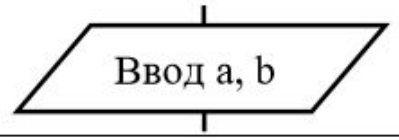
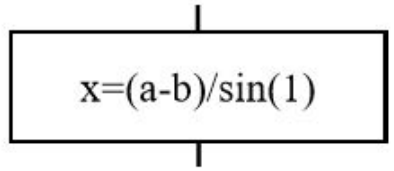
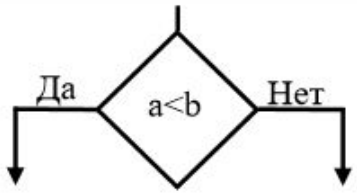
ГРАФИЧЕСКОЕ ОПИСАНИЕ АЛГОРИТМА

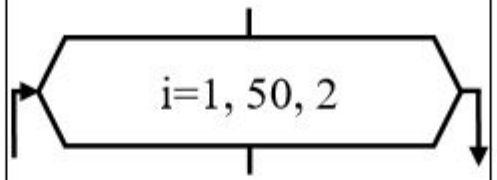
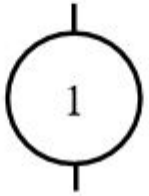
Блок-схема – описание структуры алгоритма с помощью геометрических фигур с линиями-связями, показывающими порядок выполнения отдельных инструкций.

Достоинства	Недостатки
• Наглядность; • Наиболее используемый способ.	• Может возникнуть сложность с описанием некоторых операций.

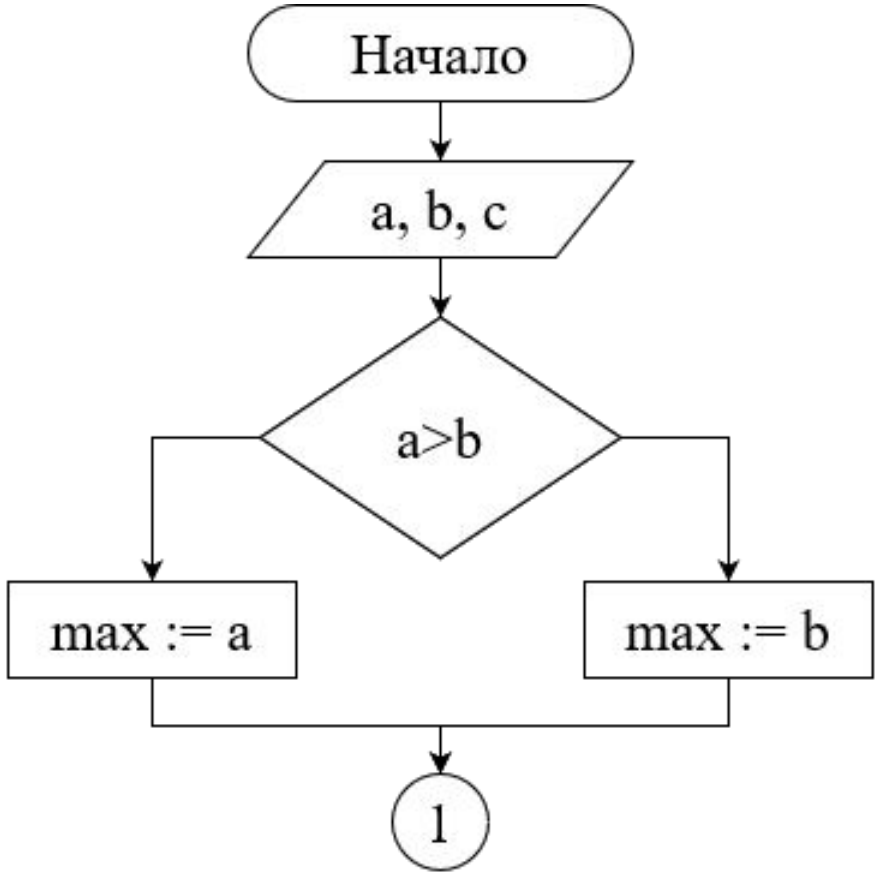
В блок-схеме каждому типу действия соответствует определенная геометрическая фигура, называемая **блочным символом**.
Блочные символы соединяются **линиями переходов**, которые определяют очередность выполнения действий.

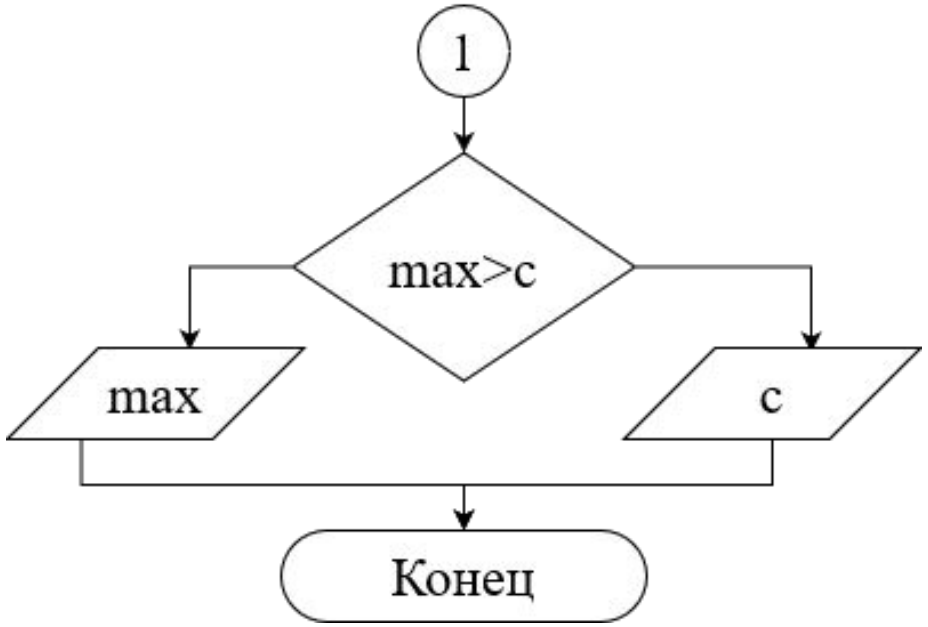
ОСНОВНЫЕ КОНСТРУКЦИИ, ИСПОЛЬЗУЮЩИЕСЯ ДЛЯ ПОСТРОЕНИЯ БЛОК-СХЕМ

Название символа	Обозначение и пример заполнения	Пояснение
Пуск-останов		Начало, конец алгоритма; вход и выход в подпрограмму
Ввод-вывод		Ввод-вывод в общем виде
Процесс		Вычислительное действие или последовательность действий
Решение (условие)		Проверка условий

Модификация		Начало цикла
Предопределенный процесс		Вычисление по подпрограмме, стандартной программе
Документ		Вывод результатов на печать
Соединитель		Соединение прерванных линий переходов

ПРИМЕР: НАЙТИ НАИБОЛЬШЕЕ ЧИСЛО ИЗ ТРЕХ ЗАДАННЫХ (A, B, C) (ГРАФИЧЕСКОЕ ОПИСАНИЕ)





ПСЕВДОКОД

Псевдокод – описание структуры алгоритма на естественном, частично формализованном языке. Представляет собой систему обозначений и правил, используемую для единообразной записи алгоритмов.

В псевдокоде не приняты строгие синтаксические записи, но используются некоторые формальные конструкции и общепринятая математическая символика.

В псевдокоде есть служебные слова смысл которых строго определен.

АЛГОРИТМИЧЕСКИЙ ЯЗЫК
СЛУЖЕБНЫЕ СЛОВА

Служебное слово	Значение
алг	алгоритм
арг	аргумент
рез	результат
нач	начало
кон	конец
цел	целый
вещ	вещественный
сим	символьный
лит	литерный
лог	логический
таб	таблица
нц	начало цикла
кц	конец цикла
длин	длина

Служебное слово	
дано	да
надо	нет
если	при
то	выбор
иначе	ввод
все	вывод
пока	утв
для	
от	
до	
знач	
и	
или	
не	

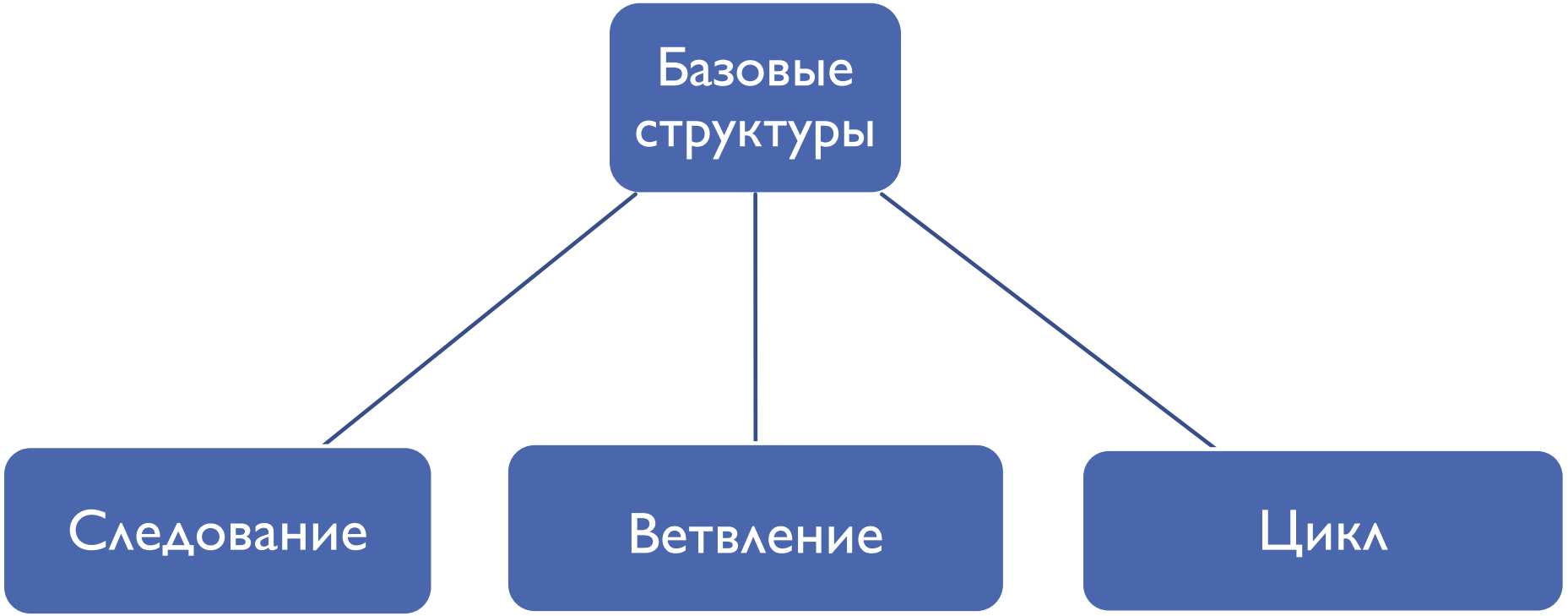
КОМАНДЫ АЛГОРИТМИЧЕСКОГО ЯЗЫКА

- Команда присваивания: A:=B
- Команды **ввода** и **вывода**:
  ввод имена_переменных;
  вывод имена_переменных, выражения, текст
- Команды **если** и **выбор**: применяются для организации ветвлений
- Команды **для** и **пока**: применяются для организации циклов

ПРИМЕР: НАЙТИ НАИБОЛЬШЕЕ ЧИСЛО ИЗ ТРЕХ ЗАДАННЫХ (A, B, C) (ПСЕВДОКОД)

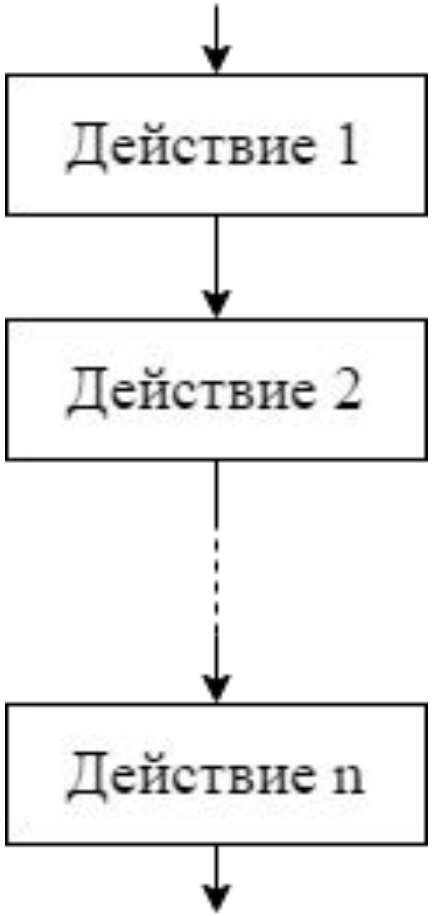
```
алг max (арг цел a,b,c, рез цел max)
нач
    ввод a,b,c
    если a>b
        то max:=a
        иначе max:=b
    все
    если max>c
        то вывод max
        иначе вывод c
    все
кон
```

БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ



БАЗОВАЯ СТРУКТУРА «СЛЕДОВАНИЕ»

Данная структура состоит из последовательно выполняющихся блоков.
Примером является стандартный процесс вычисления: ввод значений, вычисление по формуле, вывод.



<u>Алгоритмический язык:</u>
Действие 1
Действие 2
…
Действие n

БАЗОВАЯ СТРУКТУРА «ВЕТВЛЕНИЕ»

Имеет 4 формы представления. Позволяет выбрать один из альтернативных вариантов.

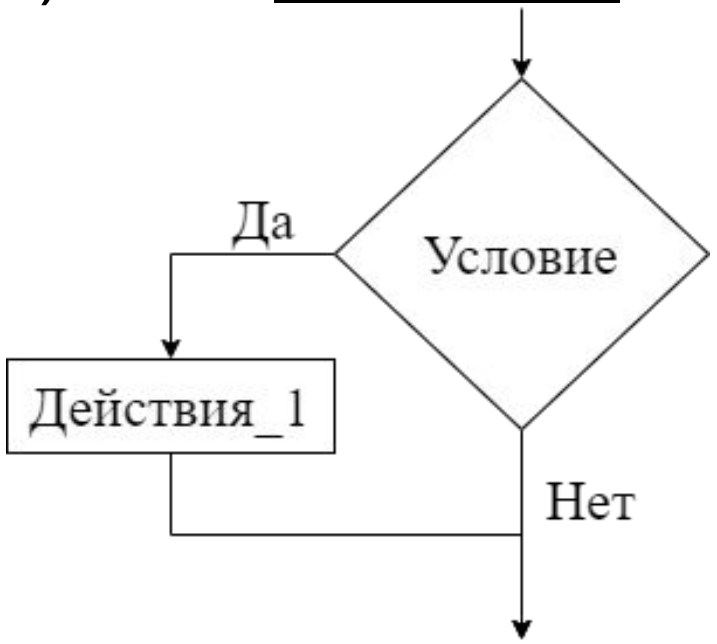
Форма 1. если-то (неполная развилка)

Если «условие» верно , тогда выполнить «действия 1», иначе ничего не выполнять

Алгоритмический язык:

если условие
то действия_1
все

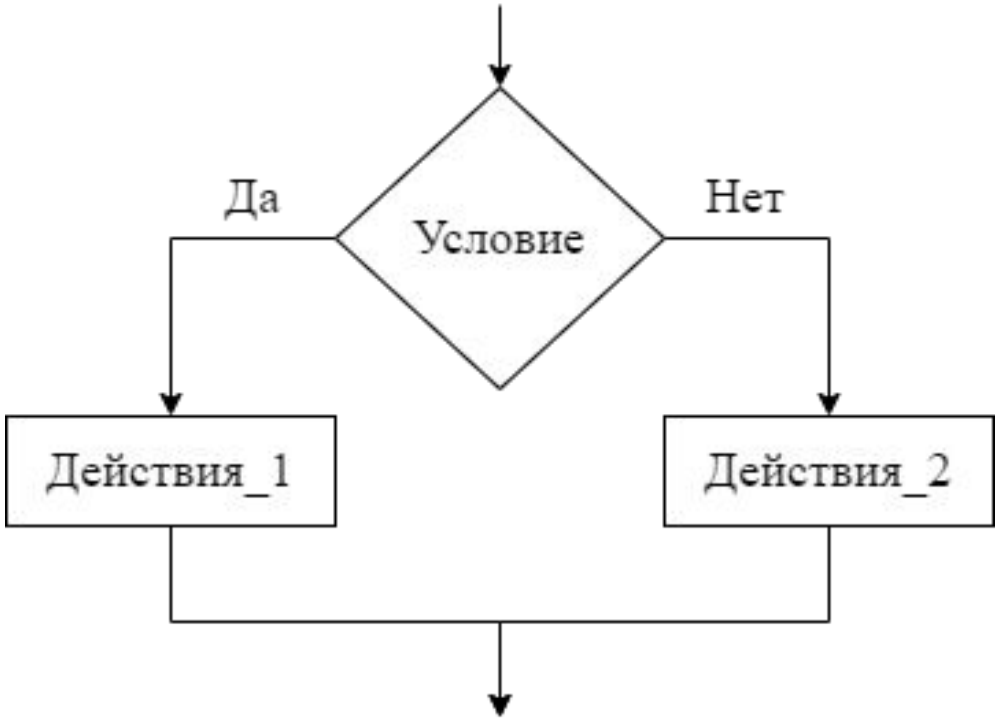
Блок-схема:



Форма 2. если-то-иначе (полная развилка)

Если «условие» верно , тогда выполнять «действия 1» (линия Да), иначе выполнять «действия 2» (линия Нет).

Блок-схема:

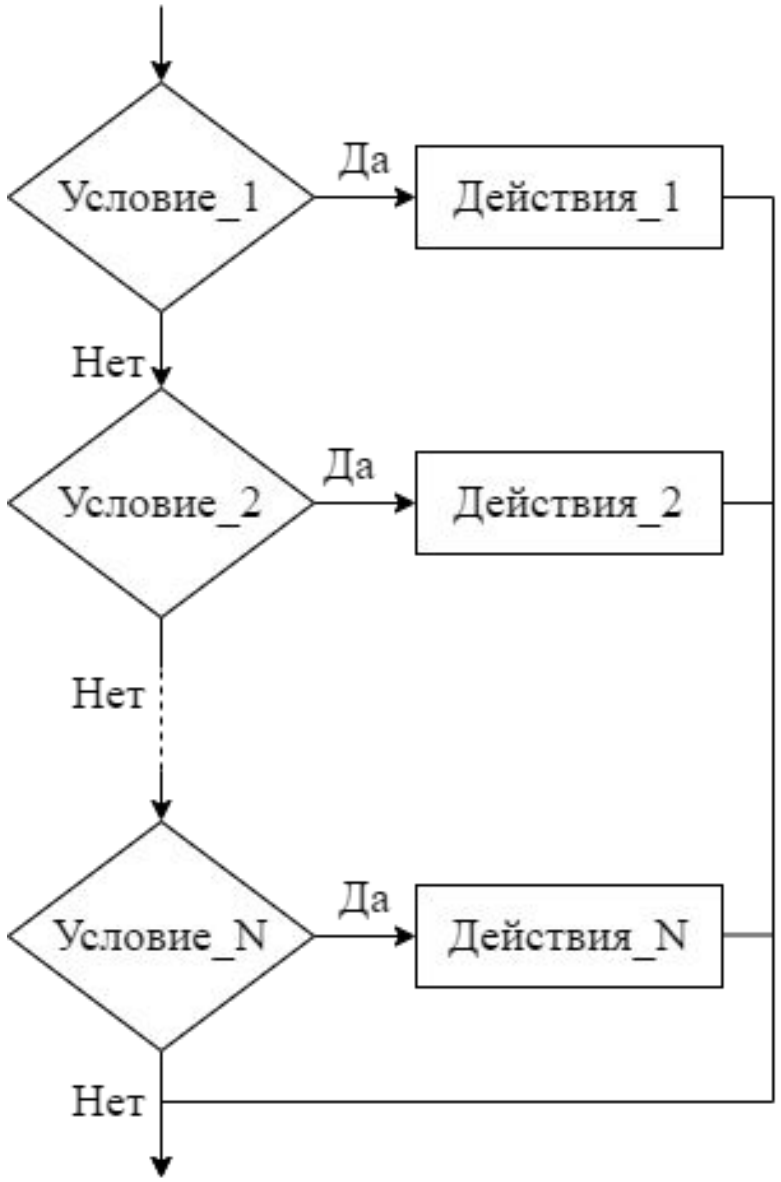


Алгоритмический язык:

если условие
    то действия_1
    иначе действия_2
все

Форма 3. Выбор

Блок-схема:

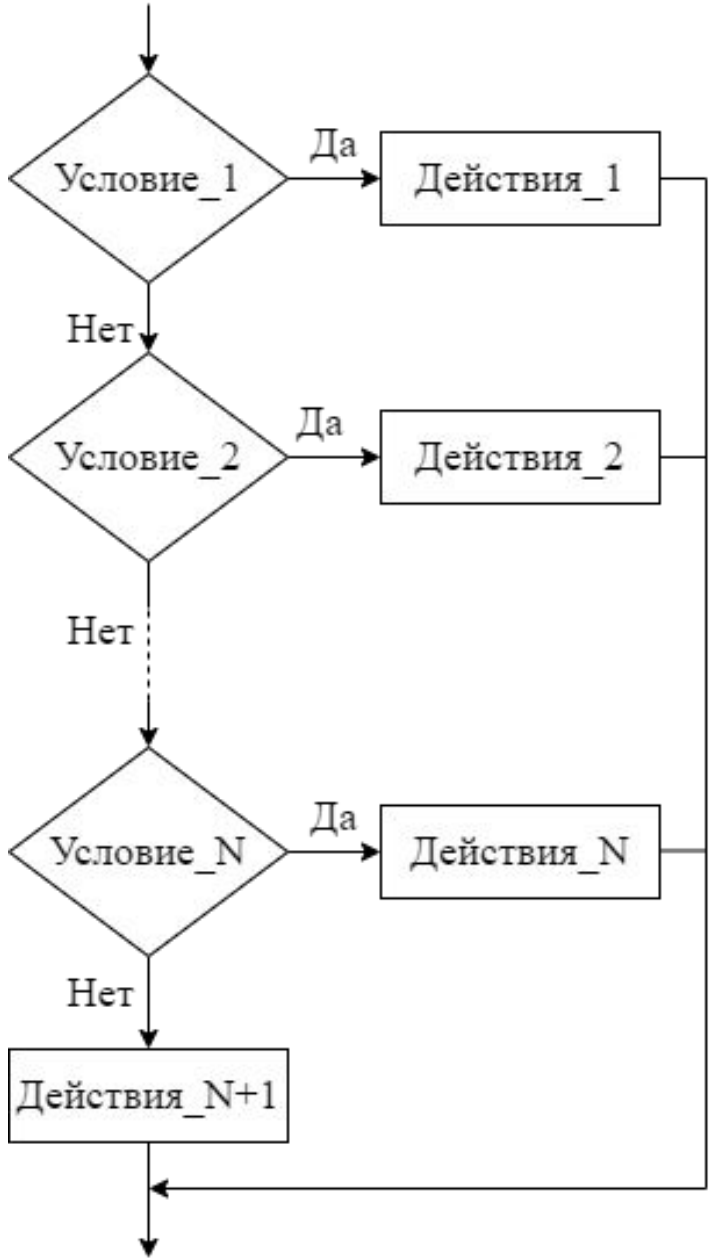


Алгоритмический язык**:**

выбор
    при условие_1: действия_1
    при условие_2: действия_2
    …
    при условие_N: действия_N
все

Форма 3. Выбор-иначе

Блок-схема:



Алгоритмический язык:

выбор
    при условие_1: действия_1
    при условие_2: действия_2
    …
    при условие_N: действия_N
    иначе действия_N+1
все

БАЗОВАЯ СТРУКТУРА «ЦИКЛ»

С помощью данной структуры выполняется одно и то же действие.
Повторение осуществляется с помощью **параметра цикла.**

Форма 1. Цикл с предусловием (цикл типа **пока**)

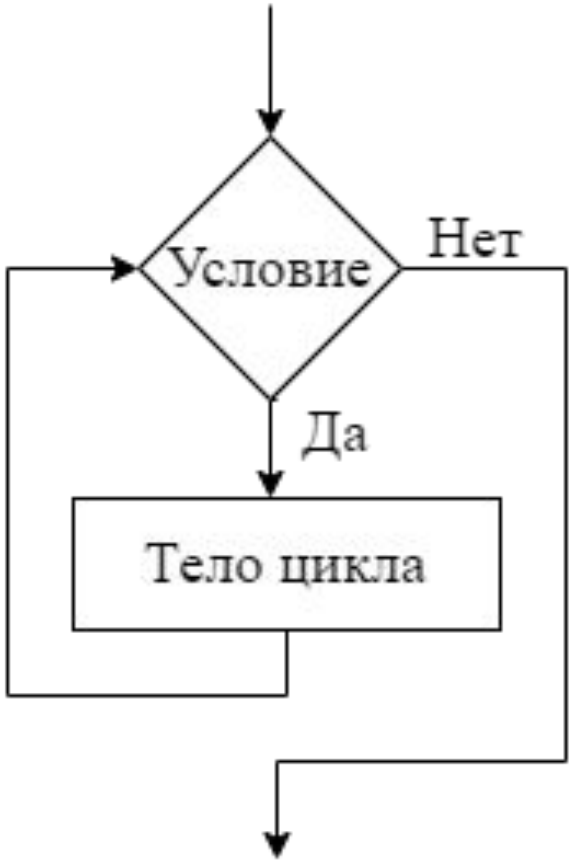
Алгоритмический язык:

нц пока условие
    тело цикла
кц

В циклах с предусловием сначала проверяется условие, если оно истинно, то выполняются команды из тела цикла.

Выполнение прекращается, когда условие становится ложным.

Блок-схема:



Форма 2. Цикл с постусловием (цикл типа **до**)

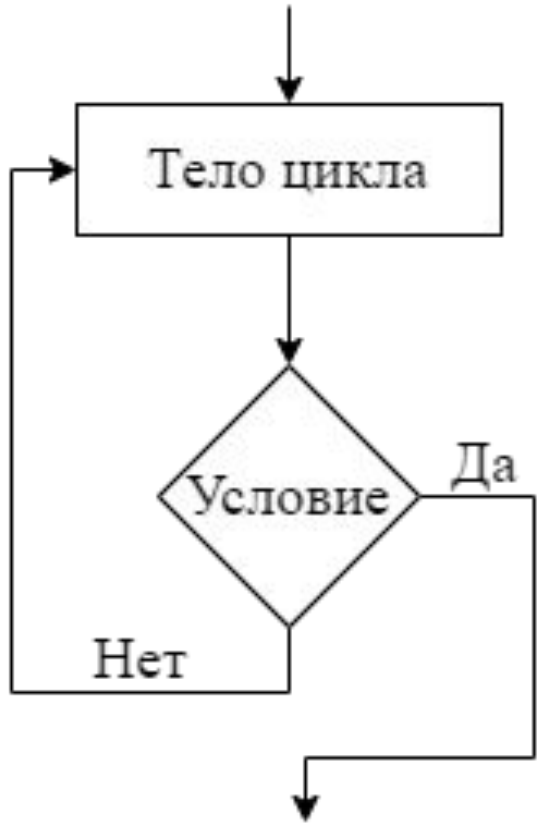
Алгоритмический язык:

нц
    тело цикла
кц при условие

В циклах с предусловием сначала выполняется действие (поэтому данный тип цикла выполняется хотя бы раз), а затем проверяется условие.

Выполнение прекращается, когда условие становится истинным.

Блок-схема:



Форма 3. Цикл с параметром (цикл типа **для**)

Алгоритмический язык:

нц для i **от** i1 **до** i2
    тело цикла
кц

Счетчику цикла **i** присваивается начальное значение **i1** и выполняется тело. После счетчик **i** увеличивается на шаг **n** и проверяется условие **i<=i2**.

Цикл завершается как только счетчик цикла **i** становится больше **i2**.

Блок-схема:

ИТЕРАЦИОННЫЕ ЦИКЛЫ

Итерационные циклы – это циклы, в которых к решению приходят путем последовательного приближения к искомому результату.

Особенностью данного цикла является то, что заранее число повторений команд неизвестно.

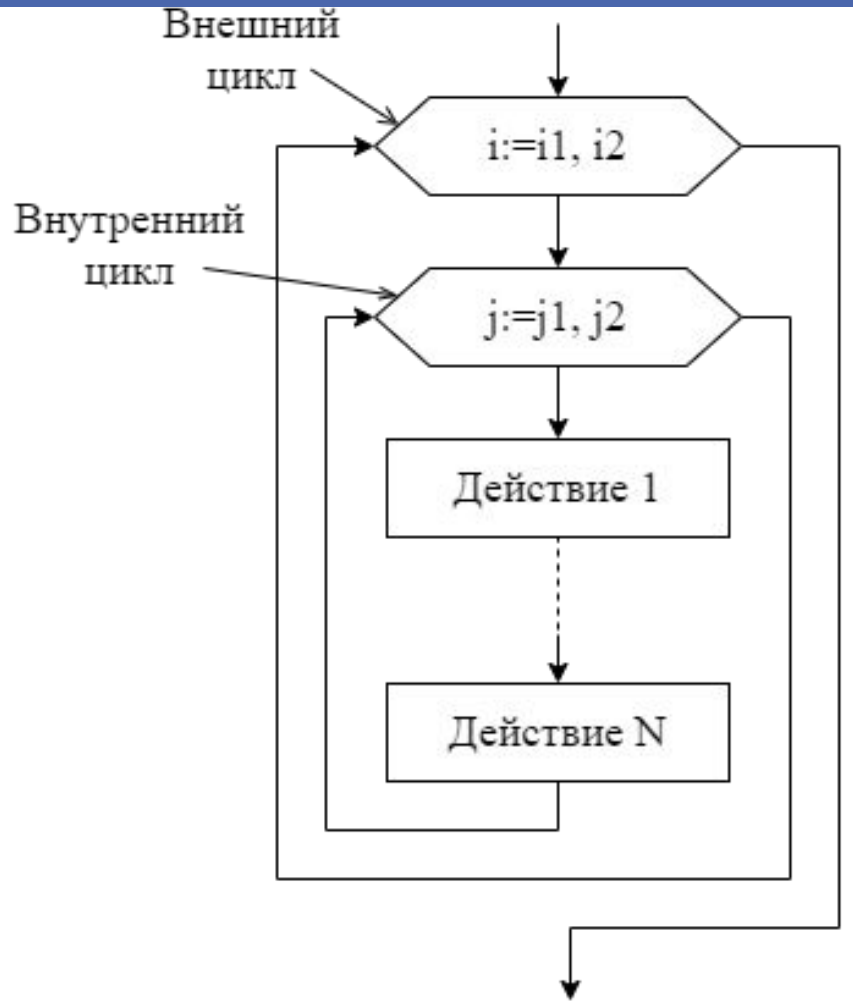
Для корректной работы необходимо обеспечивать достижение условия выхода из цикла, иначе произойдет «зацикливание».

Данные циклы используются в итерационных алгоритмах.

ВЛОЖЕННЫЕ ЦИКЛЫ

Вложенный цикл – это цикл, находящийся внутри другого цикла. Цикл, содержащий в себе другой цикл называют **внешним**, а цикл, содержащийся в теле другого цикла – **внутренним**.

Глубина вложения циклов (количество циклов вложенных в друг друга) может быть различной.



ПРИМЕР: ВЫЧИСЛИТЬ СУММУ ЭЛЕМЕНТОВ ЗАДАННОЙ МАТРИЦЫ A(5,3). (ВЛОЖЕННЫЕ ЦИКЛЫ)

Матрица A

	1	2	3
1			
2			
3			
4			
5			

Алгоритмический язык:

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

Блок-схема: