

Алгоритмические структуры.

Алгоритмы можно представлять как некоторые структуры, состоящие из отдельных базовых (т.е. основных) элементов. Естественно, что при таком подходе к алгоритмам изучение основных принципов их конструирования должно начинаться с изучения этих базовых элементов. Для их описания будем использовать язык схем алгоритмов и школьный алгоритмический язык.

Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур:
следование, ветвление, цикл.

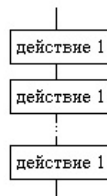
Характерной особенностью базовых структур является наличие в них одного входа и одного выхода.

1. **Базовая структура следование.** Образуется из последовательности действий, следующих одно за другим:

Алгоритмический язык

действие 1
действие 2
.....
действие n

Язык блок-схем



2. **Базовая структура ветвление.** Обеспечивает в зависимости от результата проверки условия (да или нет) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

Структура ветвление существует в четырех основных вариантах:

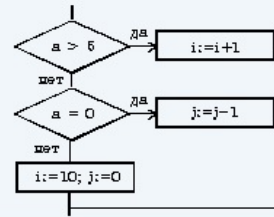
- если-то;
- если-то-иначе;
- выбор;
- выбор-иначе.

Алгоритмический язык	Язык блок-схем
<i>если-то</i>	
если условие то действия все	
<i>если-то-иначе</i>	
если условие то действия 1 иначе действия 2 все	
<i>выбор</i>	
выбор при условии 1: действия 1 при условии 2: действия 2 при условии N: действия N все	
<i>выбор-иначе</i>	
выбор при условии 1: действия 1 при условии 2: действия 2 при условии N: действия N иначе действия N+1 все	

Примеры команды если

если $x > 0$ то $y := \sin(x)$ все	
если $a > b$ то $a := 2*a; b := 1$ иначе $b := 2*b$ все	
выбор при $n = 1: y := \sin(x)$ при $n = 2: y := \cos(x)$ при $n = 3: y := 0$ все	

выбор
 при $a > 5$: $i := i+1$
 при $a = 0$: $j := j+1$
 иначе $i := 10$; $j := 0$
 все



Базовая структура цикл. Обеспечивает многократное выполнение некоторой совокупности действий, которая называется телом цикла. Основные разновидности циклов представлены в таблице:

Алгоритмический язык	Язык блок-схем
Цикл типа пока	
Предписывает выполнять тело цикла до тех пор, пока выполняется условие, записанное после слова пока.	
нц пока условие тело цикла (последовательность действий) кц	<pre> graph TD Start(()) --> D{условие} D -- да --> P[тело цикла] P --> D D -- нет --> Exit(()) </pre>
Цикл типа для	
Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне.	
нц для i от $i1$ до $i2$ тело цикла (последовательность действий) кц	<pre> graph TD Start(()) --> P[i = i1, i2] P --> C[тело цикла] C --> P </pre>
Примеры команд пока и для	
нц пока $i \leq 5$ $S := S+A[i]$ $i := i+1$ кц	<pre> graph TD Start(()) --> D{i <= 5} D -- да --> P["S := S+A[i]; i := i+1"] P --> D D -- нет --> Exit(()) </pre>
нц для i от 1 до 5 $X[i] := i*i$ $Y[i] := X[i]/2$ кц	<pre> graph TD Start(()) --> P[i = 1, 5] P --> C["X[i] := i*i; Y[i] := X[i]/2"] C --> P </pre>