

Паскаль изначально создавался по строгим канонам структурного программирования. Современные версии этого языка справедливо относят к профессиональным средствам.

Так какой же язык выбрать, спросите Вы. Да любой!



**Практика показывает, что для вступления в ряды доблестных программистов достаточно изучить и, главное, понять хотя бы один алгоритмический язык.**

После этого освоение других - уже не проблема.

*Программиста спрашивают:*



- Как вам удалось так быстро выучить английский язык?
- Да, ерунда какая. Они там почти все слова из C++ взяли.

Но изучить язык программирования - не значит вызубрить назубок все его операторы. При работе многие используют справочники и готовые примеры. Так сказать, знание со словарем. Главное - понимание. Необходимо знать возможности языка, уметь правильно применять его операторы, процедуры и функции, создать эффективный программный продукт.

## Разработка алгоритма

На первом этапе создания программы программист должен определить последовательность действий, необходимых для решения поставленной задачи. Этот этап называется разработкой алгоритма.



**Алгоритм - это точное предписание, определяющее процесс перехода от исходных данных к результату.**

Но предписание можно считать алгоритмом только тогда, когда оно обладает тремя следующими свойствами: *однозначностью, массовостью и результативностью.*

Под однозначностью алгоритма понимается единственность толкования правил выполнения действий и порядка их выполнения. Алгоритм не должен оставлять место произволу при своей реализации.

Массовость алгоритма означает возможность его применения для решения класса задач, предполагает его правильную работу при меняющихся в заданных пределах значениях исходных данных. Результативность алгоритма предполагает, что его выполнение должно приводить к получению определенного результата.

Рассмотрим, например, алгоритм вычисления действительных корней квадратного уравнения:

$$ax^2 + bx + c = 0$$

Исходные данные - это коэффициенты уравнения:  $a$  - при второй степени неизвестного;  $b$  - при первой степени неизвестного;  $c$  - при нулевой степени неизвестного.

Искомый результат - значения корней уравнения  $x_1$  и  $x_2$ .

Предписание:

1. Вычислить значение дискриминанта ( $d$ ) уравнения по формуле:

$$d = b^2 - 4ac$$

2. Если значение дискриминанта больше или равно нулю, то вычислить значения корней уравнения по формулам:

$$x_1 = \frac{-b - \sqrt{d}}{2a}; x_2 = \frac{-b + \sqrt{d}}{2a}$$

3. Если значение дискриминанта меньше нуля, то уравнение действительных корней не имеет.

Приведенное предписание обладает всеми свойствами алгоритма:

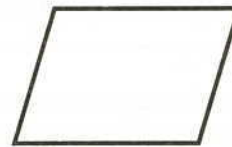
- однозначностью (в предписании указано, как обозначаются все коэффициенты уравнения, и приведены формулы для вычисления значений дискриминанта и корней уравнения);
- массовостью (в предписании указаны не конкретные значения коэффициентов, а приведены формулы, в которых использованы обозначения коэффициентов);
- результативностью (при выполнении предписания получается результат - значения корней уравнения или вывод об отсутствии решения).

Алгоритм решения задачи может быть представлен в виде словесного описания (вспомните кипение воды) или *графической блок-схемой*.

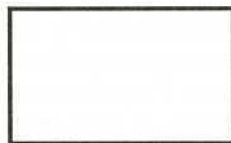
В блок-схемах для обозначения логически различных фрагментов программы используют соответствующие стандартные графические обозначения. Основные элементы блок-схем - это: Начало/Конец, Ввод/Вывод, Обработка и Выбор.



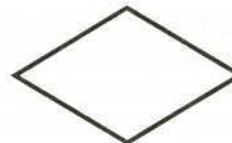
Начало/Конец



Ввод/Вывод



Обработка



Выбор