

# **Формы записей алгоритмов**

# План

- Словесная форма записи алгоритмов;
- Графическая форма записи алгоритмов;
- Псевдокоды;
- Программная форма записи алгоритмов.

- словесная (запись на естественном языке);
- Словесный способ записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке.  
Например. Записать 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:

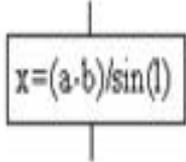
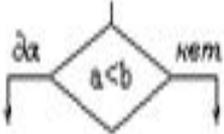
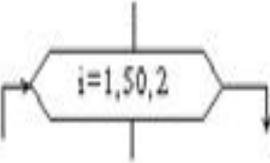
- строго не формализуемы;
- страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний.

- графическая (изображения из графических символов);

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным.

**При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий. Такое графическое представление называется схемой алгоритма или блок-схемой.**

В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде **блочного символа**. Блочные символы соединяются **линиями переходов**, определяющими очередность выполнения действий. В таблице приведены наиболее часто употребляемые символы.

Название символа	Обозначение и пример заполнения	Пояснение
<p style="text-align: center;">I</p> <p style="text-align: center;">Процесс</p>		<p style="text-align: center;">Вычислительное действие или последовательность действий</p>
<p style="text-align: center;">Решение (Условие)</p>		<p style="text-align: center;">Проверка условий</p>
<p style="text-align: center;">Модификация</p>		<p style="text-align: center;">Начало цикла</p>
<p style="text-align: center;">Предопределенный процесс</p>		<p style="text-align: center;">Вычисления по подпрограмме, стандартной подпрограмме</p>

<p>Ввод-вывод</p>		<p>Ввод-вывод в общем виде</p>
<p>Пуск-останов</p>		<p>Начало, конец алгоритма, вход и выход в подпрограмму</p>
<p>Документ</p>		<p>Вывод результатов на печать</p>

- псевдокоды (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов. Псевдокод занимает промежуточное место между естественным и формальным языками.  
С одной стороны, он близок к обычному естественному языку, поэтому алгоритмы могут на нем записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются некоторые формальные конструкции и математическая символика, что приближает запись алгоритма к общепринятой математической записи.

- **В псевдокоде не приняты строгие синтаксические правила для записи команд.** Однако в псевдокоде обычно имеются **некоторые конструкции, присущие формальным языкам**, что облегчает переход от записи на псевдокоде к записи алгоритма на формальном языке.

В частности, в псевдокоде, так же, как и в формальных языках, есть **служебные слова**. Они выделяются в печатном тексте жирным шрифтом, а в рукописном тексте подчеркиваются.

Примером псевдокода является школьный алгоритмический язык в русской нотации (**школьный А-Я**), описанный в учебнике А.Г. Кушниренко и др.

Этот язык в дальнейшем мы будем называть просто "алгоритмический язык".

Текстовая форма записи алгоритма (*псевдокод*) – шаги алгоритма и последовательность их выполнения задаются набором ключевых слов.

- **нач** начало программы
- **кон** конец программы
- **если-то-иначе** проверка условия
- **ВВОД** ввод данных
- **ВЫВОД** вывод данных
- **для-от-до-нц-кц** цикл со счетчиком (нц – начало цикла, кц – конец)
- **пока-нц-кц** цикл с предусловием
- **нц-кц-пока** цикл с постусловием

- **программная** (тексты на языках программирования).
- При записи алгоритма в словесной форме, в виде блок-схемы или на псевдокоде допускается определенный произвол при изображении команд. Вместе с тем такая запись точна настолько, что позволяет человеку понять суть дела и исполнить алгоритм.
- Однако на практике в качестве исполнителей алгоритмов используются специальные автоматы — компьютеры. Поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на понятном ему языке. И здесь на первый план выдвигается необходимость точной записи команд, не оставляющей места для произвольного толкования их исполнителем.
- Следовательно, **язык для записи алгоритмов должен быть формализован**. Такой язык принято называть **языком программирования**, а запись алгоритма на этом языке — **программой для компьютера**.

# Программа-

это *реализация* алгоритма на конкретном языке программирования.

Совокупность существующих программ образует **программное обеспечение** (ПО)

# ПО принято делить на 2 вида:

- **Системное ПО** обеспечивает работу компьютера и внешних устройств, а также поддержку прикладных программ;
- **Прикладное ПО** предназначено для решения конкретных задач пользователя.

# Этапы разработки программы

1. Определение входных и выходных данных, требований к программе;
2. Разработка алгоритма;
3. Кодирование (программирование);
4. Компиляция и [отладка](#);
5. *Тестирование* – проверка правильности работы программы на наборах тестовых данных с заранее известным результатом;
6. *Документирование и поддержка* – создание справочной системы и документации к программе, возможно, расширение ее функциональности, выпуск новых версий, исправление ошибок, которые практически неизбежны в любой сложной программной системе.

# Отладка -

Поиск ошибок в программе.

Возможны программные ошибки 3-х видов:

- *синтаксические* (ошибки в правилах языка);
- *алгоритмические* (ошибки в логике программы);
- *ошибки времени исполнения.*

# Компиляция -

- это процесс преобразования программы в машинный код.