

Алгоритмы

Алгоритм — некоторое предписание действий исполнителю, строгое выполнение которых приводит к решению поставленной задачи за конечное число шагов.

- *тенденции последних лет в развитии курса информатики таковы, что **алгоритм** рассматривается не просто как план действий, предписание исполнителю, но более широко — **как модель деятельности.***

Примеры алгоритмов из ЖИЗНИ:

- правило перехода улицы,
- рецепт пирога,
- инструкция по изготовлению яда и т.д.

Задание 1. Что из нижеперечисленного является алгоритмом, а что нет?

- 1) правила игры в футбол;
- 2) политическая карта мира;
- 3) телефонный справочник;
- 4) файл, содержащий информацию об установке программы;
- 5) вычисление корней квадратного уравнения;
- 6) приготовление завтрака;
- 7) решение шахматной задачи;
- 8) инструкция по пользованию видеомаягнитофоном

Свойства алгоритма



✓ **Понятность** – исполнитель алгоритма должен знать, как его выполнять

✓ **Определённость** – каждое правило алгоритма должно быть чётким, однозначным и не оставлять места для произвола

✓ **Результативность** – состоит в том, что алгоритм должен приводить к решению задачи за конечное число шагов

✓ **Дискретность** – алгоритм должен представлять процесс решения задачи как последовательное выполнение простых шагов.

✓ **Массовость** – означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными.

Задание 2. Придумайте систему команд для исполнителя алгоритма чистки картошки, заварки чая.

Задание 3. Предложите свой алгоритм действий для достижения следующего результата – не пойти завтра в школу.

Задание 4. В приведенных ниже последовательностях каждый следующий элемент получен по некоторому строгому алгоритму. Разгадав его, продолжите ряд:

- а,в,д,ё,з,й;
- 1,2,4,8,16,32;
- 1,4,9,16,25;
- Победа, обеда, беда, еда;

Виды алгоритмов. Базовые структуры алгоритмов: следование, ветвление, повторение.

Алгоритмы бывают трех основных видов, которые и являются базовыми при написании программ.

- Первый тип – *линейный* алгоритм; такой, в котором все действия выполняются в строгом порядке, последовательно, одно за другим. Типичный жизненный пример – рецепт пирога.
- Второй тип – *разветвляющийся* алгоритм; такой, в котором выполняются те или иные действия в зависимости от выполнения или невыполнения некоего условия. Пример из жизни – правило перехода улицы по светофору. Если горит красный – стоим, если горит зеленый – идем.
- Третий тип – *циклический* алгоритм; такой, в котором присутствуют повторяющиеся действия с какой-либо изменяющейся величиной, так называемым параметром. Пример – колка дров. Берем полено – ставим его, колем топором, берем второе полено и т.д., пока поленья не закончатся, и эта работа нам не надоест.

Приведите свои примеры

- линейного,
- разветвляющегося,
- циклического алгоритмов.

Способы записей алгоритмов

- Словесный способ – представляет собой описание последовательных этапов обработки данных. Алгоритм задаётся в произвольном изложении на естественном языке. Словесный способ не имеет широкого распространения, т.к. такие описания страдают многословностью записей и допускают неоднозначности толкования отдельных предписаний.
- Графический способ – изображения из графических символов (блок-схемы)
- Псевдокоды (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.)

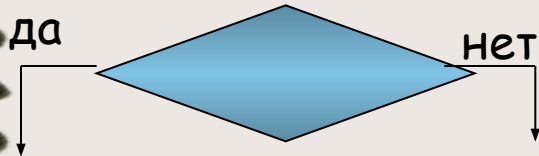
Графический способ представления алгоритмов (блок-схема).

- Графический способ является вспомогательным. Графическое изображение алгоритма позволяет наглядно проследить и осмыслить процесс выполнения алгоритма.
- Используемые геометрические фигуры соответствуют одному шагу алгоритма и имеют стандартный смысл.

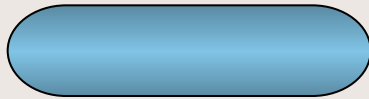
Блочные символы



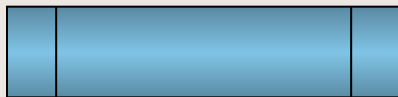
Вычислительное действие или
последовательность действий



Проверка условий



Начало, конец алгоритма, вход и выход в
подпрограмму



Вычисления по подпрограмме

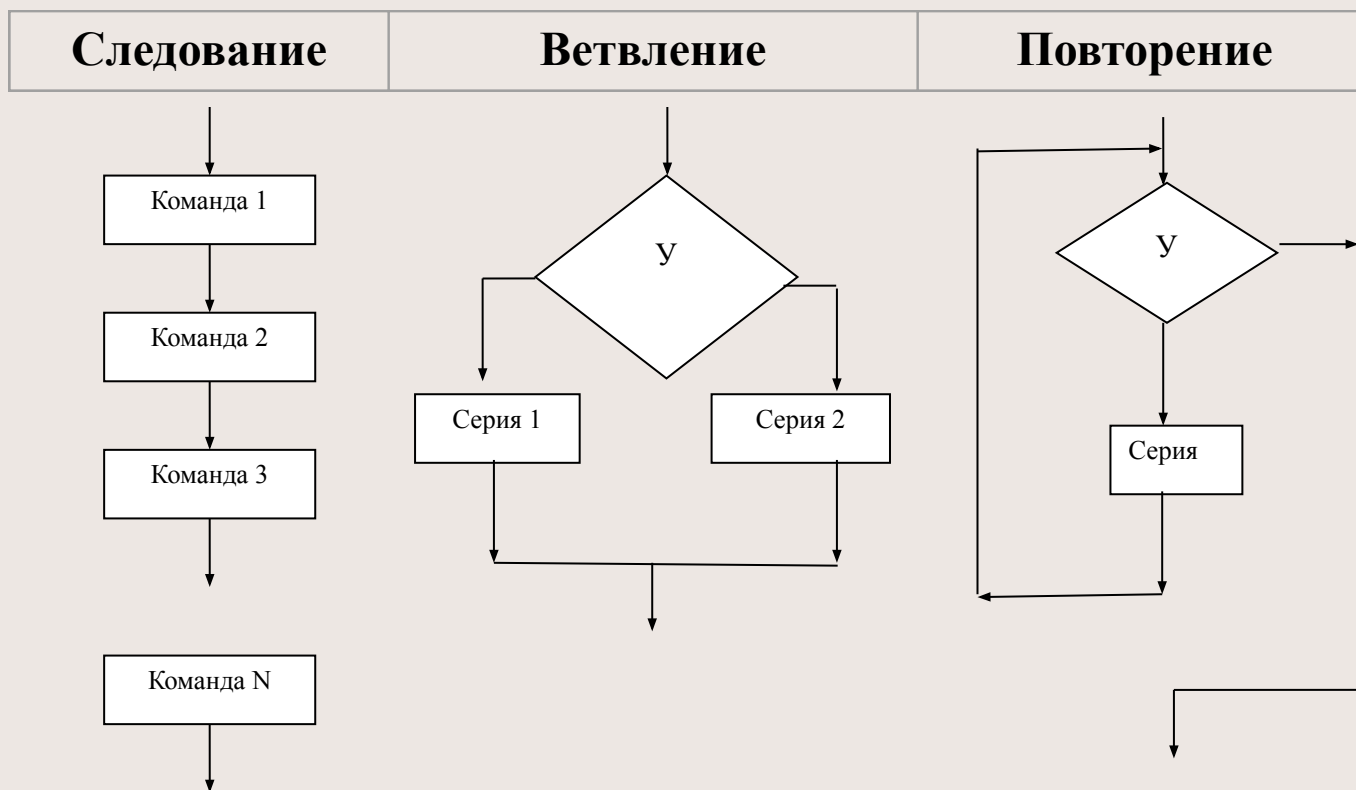


Ввод-вывод в общем виде



Вывод результатов на печать

Основные структуры алгоритмов на языке блок-схем



Алгоритмический язык

- *Алгоритмический язык* – это система обозначений и правил для единообразной записи алгоритмов и их исполнения.
- *Система обозначений* алгоритмического языка состоит из служебных слов, математической символики и функций, которые делают запись алгоритма наглядной и единообразной.

Служебные слова

алгоритмического языка

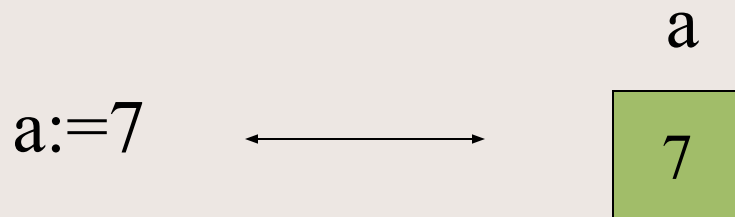
- алг (алгоритм)
- арг (аргумент)
- вещ (вещественный)
- все
- до
- если
- знач (значение)
- и
- или
- иначе
- кон (конец)
- кц (конец цикла)
- лит (литерный)
- лог (логический)
- нат (натуральный)
- нач (начало)
- нц (начало цикла)
- от
- пока
- рез (результат)
- таб (таблица)
- то
- цел (целый)
- шаг

- *Величины* – данные, постоянные и переменные.
- *Данные* – необработанная информация.
- *Постоянные* – имена фиксированных значений.
- *Переменная* – это область памяти с уникальным именем, где можно сохранить или откуда можно извлечь определенное значение.

Величина имеет три важных характеристики:

Имя	Тип	Множество значений
Слово, начинающееся с буквы (кроме служебных слов)	<u>нат</u> <u>цел</u> <u>вещ</u> <u>лог</u> <u>лит</u>	натуральные целые действительные «истина» или «ложь» «текст»

- К моменту использования переменной величины в алгоритме она должна получить некоторое конкретное значение соответствующего типа. Задать конкретное значение величине можно с помощью команды *присваивания*.
- ***Команда присваивания*** ($:=$) предназначена для записи значения величины в область памяти с определенным именем.



Общий вид алгоритма на учебном алгоритмическом языке

алг название алгоритма (описание формальных параметров)

арг условие применимости алгоритма

рез цель выполнения алгоритма

нач описание неформальных параметров

тело алгоритма

кон