**Формы представления алгоритмов**

1. Словесная форма представления
   1. Словесная форма записи не широко распространена но-за многословности и отсутствия наглядности.

Преимущества: простой способ описания действий

Недостатки: описание строго не формализуется; запись получилась многословной, отдельные предписания (действия) допуская неоднозначность толкования.

1. Табличная форма представления
   1. Представляется в виде «чек-лист»’а

Преимущества: организованный список

Недостатки: может быть сложным в больших списках

1. Графовые (блок-схемы) формы представления
   1. Основной способ представления алгоритмов

Преимущества: стандартизирован, удобен, не сильно громозкий

Недостатки:

1. Псевдокод
   1. Краткое описание кода, понятное человеку

Преимущества: Довольно удобно и компактно описывает код на понятном человеку языке

Недостатки:

**Программы**

* **Программа** – формальная запись алгоритма на одном из языков программирования.
* Языки программирования характеризуются строго определённым синтаксисом, то есть свободное толкование конструкций не допускается.
* **Компьютерная программа** — последовательность инструкций, предназначенная для исполнения устройством управления вычислительной машины.

**Базовы структуры алгоритма**

* Для решения прикладной задачи всегда можно составить несколько разных алгоритмов.
* Чтобы оценить насколько «хорош» алгоритм, анализируются следующие характеристики:
  + простота и легкость понимания алгоритма,
  + скорость выполнения и требуемый объём памяти.
* Наиболее важными в настоящее время являются простота и легкость понимания. Для того чтобы алгоритм был простым и легко понимаемым, он должен быть структурным и его рекомендуется строить, используя 3 основных структуры
  + Последовательность (последовательное соединение)
  + Условие (проверка условия).
    - Условие с двумя вариантами действий.
    - Условие с одним вариантом действий.
  + Цикл (циклическое повторение).
    - Цикл с предусловием.
    - Цикл с постусловием.

**Примечание**

* Во всех трёх случаях, во всех стандартных структурах блоки А и В могут в свою очередь состоять из таких же стандартных структур. Т.е. одна структура может быть вложена в другую произвольное количество раз.

**Структурный алгоритм, программа, структурное программирование, теорема о структурных алгоритма**

* Структурный алгоритм – это алгоритм, в котором используются только стандартные структуры – последовательность, условие, цикл, а также ввод-вывод.
* Структурная программа – это программа, которая точно соответствует структурному алгоритму.
* Структурное программирование – процесс разработки структурных алгоритмов и программ.
* Справедлива следующая теорема о структурных алгоритмах: для каждого алгоритма существует эквивалентный ему структурный алгоритм.

**Безусловный переход**

* В некоторых языках программирования отсутствуют полноценные структуры для реализации условий и циклов. Поэтому в дополнение к трём стандартным структурам иногда применяется безусловный переход. При выполнении безусловного перехода управление передается на указанный шаг (пункт) в алгоритме

**Ограничение использования безусловного перехода**

* Использование безусловного перехода приводит к нарушению принципов структурного программирования. Поэтому его применение в структурном программировании – ограничено. Однако существуют случаи, когда без него не обойтись, например при обработке критических ошибок. В других ситуациях хороший стиль программирования полностью исключает использование этой конструкции. Считается плохим тоном писать структурные программы с использованием безусловного перехода. Следует также отметить, что в языках машинного (низкого) уровня типа Ассемблер без этих конструкций обойтись невозможно.

**Преобразование неструктурных алгоритмов в структурные**

* Структурный алгоритм не всегда удается построить сразу. Поэтому в таких случаях можно поступать следующим образом:
  + Сначала построить неструктурный алгоритм, решающий поставленную задачу;
  + Затем с помощью специальных приемов, преобразовать его в структурный.
    - Дублирование отдельных блоков, Объединение условий, Использование переменной флажка.
* Таких приемов существует много, рассмотрим некоторые из них: дублирование отдельных блоков схемы, объединение нескольких простых условий в одно сложное, использование переменной флажка.