

# Основы алгоритмизации и программирования.

План лекции:

1. Понятие алгоритма. Исполнитель алгоритмов. Свойства алгоритма.
2. Формы представления алгоритмов.
3. Классификация алгоритмов.

## Понятие алгоритма. Исполнитель алгоритмов.

**Алгоритм** (от имени *Мухаммед ибн Мусса аль Хорезми*) – это строгая последовательность действий, выполнение которой приводит к решению поставленной задачи.

В связи с необходимостью выполнить эти действия возникает понятие «исполнитель алгоритма».

**Исполнитель** – это человек или автомат, умеющий выполнять некоторый вполне определенный набор действий. Для его характеристики используют ряд понятий:

- **среда** – место действия исполнителя;
- **система команд** – строго заданный список команд, понятных исполнителю;
- **элементарные действия** – действия, которые может выполнить исполнитель;
- **отказы** возникают, если команда вызывается при недопустимом для нее состоянии среды.

В информатике универсальным исполнителем является компьютер.

## Свойства алгоритма.

При всем многообразии алгоритмов выделяют свойства алгоритмов, общие для всех:

- **понятность для исполнителя** означает, что исполнитель должен знать, как его выполнить;
- **дискретность** означает упорядоченную последовательность четко разделенных друг от друга действий;
- **детерминированность (определенность)** означает, что результат выполнения последовательности действий однозначно определяется исходными данными;
- **массовость** означает возможность выполнения алгоритма для различных допустимых комбинаций исходных данных;
- **результативность** означает получение результата за конечное число шагов.

## Формы представления алгоритмов

Существуют следующие формы представления алгоритмов:

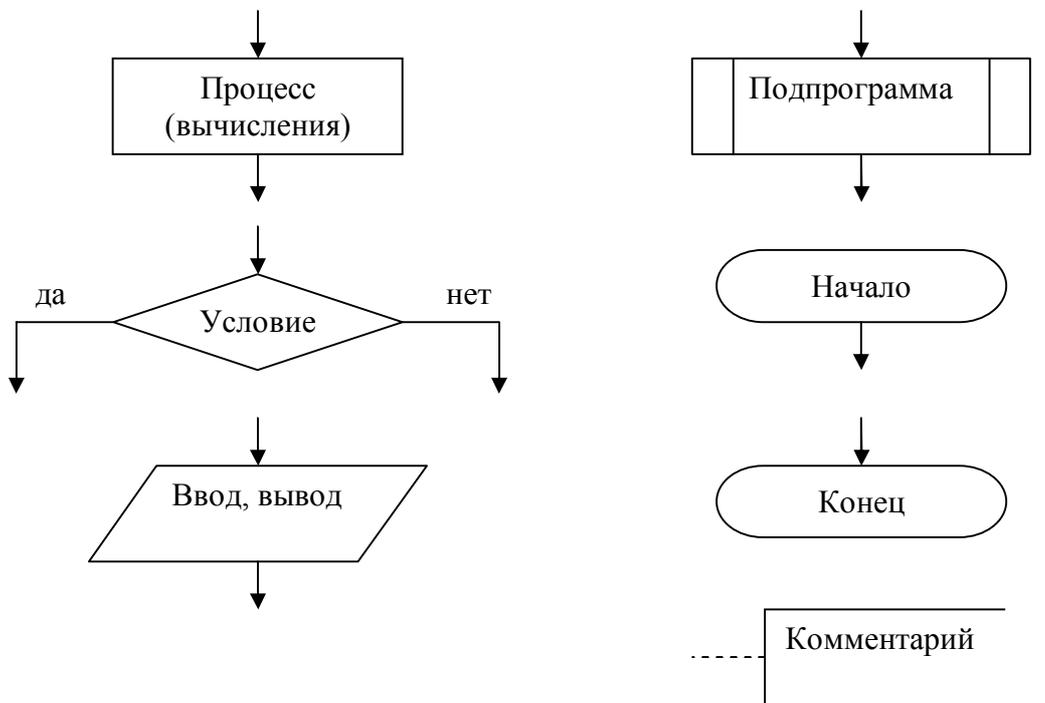
- **на естественном языке**, когда не требуется детальных разъяснений и полной детализации.

Достоинство – доступность для понимания его любым человеком.

Недостаток – в громоздкости записи, отсутствии наглядности, неточности и многозначности

- **на псевдокоде**. Псевдокод – это полужформализованное описание алгоритма на некотором условном алгоритмическом языке. Занимает промежуточное положение между естественным и формальным языком.

- **графические в виде блок-схем**, когда отдельные действия алгоритма изображают при помощи различных геометрических фигур (блоков) и связей между ними, которые обозначаются стрелками. Отличаются наглядностью, использованием графических символов, математических записей и записей на естественном языке.

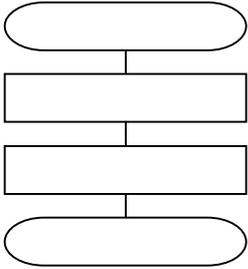
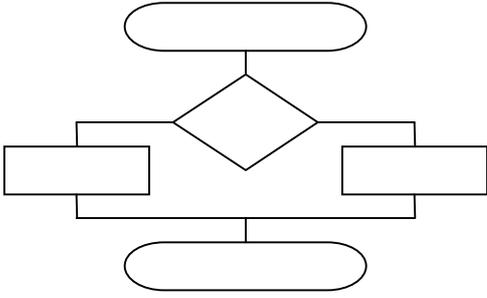
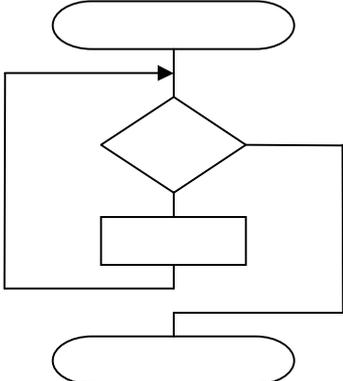


- **на языках программирования**. Языком программирования называется (формальный) язык, предназначенный для записи программ, исполняемых на ЭВМ.

..

### Классификация алгоритмов

Все существующие алгоритмы делятся на три вида: линейные, ветвящиеся и циклические.

	<p><b>Линейный алгоритм</b> при каждом исполнении предписывает однократное выполнение всех действий алгоритма в определенной последовательности.</p>
	<p><b>Ветвящийся алгоритм</b> описывает несколько возможных последовательностей действий и при каждом исполнении предписывает выполнение одной из последовательностей действий в зависимости от определенных условий.</p>
	<p><b>Циклический алгоритм</b> при каждом исполнении предписывает многократное выполнение одной и той же последовательности действий.</p>

По степени детализации алгоритмы подразделяются на укрупненные и детальные.

Укрупненный алгоритм содержит только наиболее важные (существенные) действия, отражающие суть процесса решения требуемой задачи, т.е. описывает алгоритм решения задачи обобщенно, на уровне метода решения.

Детальный алгоритм точно описывает процесс решения задачи, поэтому его предписания легко могут быть преобразованы в исполняемую программу.

Одним из приемов разработки алгоритмов является **метод пошаговой детализации (нисходящее проектирование программы, программирование сверху вниз)**. Он заключается в том, что первоначально формируется общая структура алгоритма, а дальше прорабатываются (детализируются) его отдельные блоки. Таким образом, на каждом шаге получается более простая задача. Полностью закончив детализацию, получим алгоритм решения задачи в целом.

# ЯЗЫК ТУРБО ПАСКАЛЬ

В алфавит языка входят:

1. Латинские строчные и прописные буквы: A,B,...,Z, a,b,...,z (однако при написании программы регистр не имеет значения).
2. Символ подчеркивания «\_»: my\_var
3. Пробел « » служит для разделения слов.
4. Цифры 0...9.
5. Специальные символы, участвующие в построении конструкций языка  
+ - \* / = < > [ ] . , ( ) : ; ^ @ { } \$ #

## Структура программ

**PROGRAM** Имя\_программы;

**USES** Список используемых модулей;

**LABEL** Список меток в основном блоке программы;

**CONST** Определение констант;

**TYPE** Описание типов;

**VAR** Определение глобальных переменных программы;

**ОПРЕДЕЛЕНИЕ ПРОЦЕДУР** (заголовки и тела процедур);

**ОПРЕДЕЛЕНИЕ ФУНКЦИЙ** (заголовки и тела функций);

**BEGIN**

Основной блок программы

**END.**

В действительности используются далеко не все разделы. Так самая короткая программа имеет вид:

**BEGIN**

**END.**

Однако раздел описания переменных **VAR** присутствует практически всегда. После точки, завершающей основной блок программы любой текст игнорируется (не воспринимается). Все комментарии заключаются в фигурные скобки {.....} либо (\* .... \*) и не воспринимаются компилятором.