

Введение в программирование



Turbo Pascal

9 класс

Тема 1: Введение в программирование

- Какой язык понимает процессор?
- Процессор понимает язык электрических сигналов. Он не различает сильный или слабый сигнал. Процессор принимает одно из двух состояний: **есть сигнал** (1) или **нет сигнала** (0).
- Процессор отличает **сочетания** сигналов в единицу времени как много или в какой последовательности будет скачок тока или нет.



Введение в Turbo Pascal

- **Компьютер** – это сложная вычислительная система, предназначенная для реализации процесса автоматизированной обработки данных в соответствии с задаваемыми алгоритмами.
- **Процессор** – программно-управляемое электронное устройство, построенное на одной или нескольких микросхемах. Основная функция – выполнение арифметических, логических и других операций, закодированных в командах; управление основными узлами компьютера и координация их взаимодействия.
- **Микросхема** – электронная схема, реализованная в виде полупроводниковых кристаллов.



Введение в Turbo Pascal

- **Сигнал** – условный знак для передачи на расстояние каких-нибудь сведений, сообщений.
- **Аналоговый сигнал** – сигнал, который может принимать более двух комбинаций значений.
- **Двоичный сигнал** - сигнал, который может принимать только два варианта значений.
- **Код** – это совокупность правил обозначения элементов некоторого исходного множества комбинациями символов заданного алфавита (кодowymi комбинациями).



Как осуществляется взаимодействие ПК и человека?

- **Машинная программа** – последовательность машинных команд, задающая алгоритм решения задачи в виде последовательности машинных операций.
- **Языки программирования** – система обозначений и правил для описания алгоритмов и структур данных.
- **Языки низкого уровня** – это средство записи инструкций компьютеру простыми приказами-командами на машинном уровне, ориентированном на конкретную ЭВМ.
- **Машинный язык** – управляющий код для конкретной машинной операции, определяющий, откуда взять исходные данные и куда поместить результаты выполнения операции.
- **Язык программирования высокого уровня** - это средство записи инструкций системой обозначений, близкой к человеку. Ориентирован на любую ЭВМ, т.к. в состав языка входит программа, автоматически переводящая текст на язык ЭВМ.

Тема 2: Система программирования

- **Язык программирования** должен понимать текст программы, написанный человеком, анализировать его, указывая на ошибки, переводить на машинный язык для выполнения команд процессором.
- **Система программирования** – совокупность программ для создания других программ.
- **Среда программирования** – программная оболочка для системы программирования, организующая удобный интерфейс для написания программ.



- Текст программы пишется во строенном в среду программирования текстовом редакторе. В нем можно производить простые манипуляций с текстом. Встроенный в среду программирования редактор «знает» **алфавит** и **синтаксис** данного языка, что способствует отладки текста уже на первых шагах.



Система программирования

- **Алфавит** – конечный набор знаков, состоящий из букв, десятичных и шестнадцатеричных цифр, специальных символов.
- **Синтаксис** – правила написания текста, в том числе порядок слов и знаков препинания.



- **Как текст программы понимает процессор?**
- Для этого имеются в системе программирования другие программы: **трансляторы.**
- **Любое ли действие можно запрограммировать?**
- Любое, но в каждом языке имеется стандартный набор команд, связанных с тем или иным действием.
- **Где хранятся сведения о том, какие действия нужно выполнить при той или иной команде в тексте?**
- В специальных **библиотеках программ**, т.е. в файлах встроенных в систему программирования.



- **Транслятор** – специальная программа, осуществляющая перевод созданной программы на машинный язык.
- **Интерпретатор** – транслятор, производящий покомандную обработку и выполнение исходной программы.
- **Компилятор** – преобразует всю программу в модуль на машинном языке, после этого программа записывается в память компьютера и лишь потом исполняется.



Система программирования

- **Модули** – части целого.
- **Библиотеки программ** – файлы, которые содержат информацию, подключаемую программным способом для использования в исполняемой части по мере необходимости.



Система

программирования

- Если в модулях библиотеки нет команды, которая нужна при создании программы?
- Её нужно написать на языке программирования, т.е. создать дополнительный модуль с новыми командами для данной системы программирования.
- Сколько модулей (файлов) входят в среду программирования?
- Много, но не всегда все модули используются, объем программного обеспечения языка может меняться.

Необходимые модули Turbo Pascal.

- **Turbo.exe** – интегрированная среда программирования;
- **Turbo.hlp** – файл, содержащий данные для оперативной подсказки;
- **Turbo.tp** – файл конфигурации системы;
- **Turbo.tpl** – библиотека стандартных модулей TP;
- **Graph.tpu** и **Graph.int**– модули с графическими процедурами;
- **CRT.tpu** и **Crt.int** – модуль управления дисплеем и клавиатурой;
- **Egavga.bgi** – драйвер видеосистемы монитора.
- **Шрифты** – файлы с расширением **.chr**



Тема 3: Азбука языка Pascal

- **Какие существуют правила написания команд?**
- Каждая команда предназначена для конкретной цели, поэтому в ее состав могут входить как стандартные команды с аргументами, так и пользовательские, но тоже с соблюдением правил алфавита и синтаксиса.
- **Можно ли использовать буквы отличные от латинского алфавита?**
- Да, но только не в качестве команд, а только как **комментарий**, поясняющий либо текст программы, либо действия пользователя. Комментарии размещаются в специальных символах: например, в апострофах ('), фигурных скобках {}.



Азбука языка Pascal

- **Легко ли пользователю понять текст (листинг) чужой программы?**
- Легко, если он знает английский язык, т.к. из него взяты имена команд. Но каждому даже программисту необходимы справочные пособия по работе в определенной незнакомой программе.
- **Как одна команда отделяется от другой?**
- В языке TP команда заканчивается знаком `<;>`.
- В других языках команду записывают с новой строки.



Азбука языка Pascal

- **Символ** – условный знак, обозначающий букву, цифру и т.п.
- **Буква** – символ из азбуки человеческого языка.
- **Зарезервированные слова** – слова, являющиеся составной частью языка, имеют фиксированное начертание и, раз и навсегда, определенный смысл.
- **Азбука** – совокупность символов, принятых в данной письменности, располагаемых в установленном порядке, алфавит.
- **Синтаксис** – система языковых категорий, относящихся к соединениям слов и строению предложений.
- **Пунктуация** – расстановка знаков препинания.
- Какими словами пишется текст программы?



Азбука языка Pascal

- **Аргументы** – части выражений, являющиеся объектами, которые в отдельности используются командой, в состав которой они входят.
- **Выражение** – словосочетание, состоящее из слов и символов и задающее правило вычисления некоторого значения.
- **Комментарии** - текст, который помещается в программу для лучшего понимания алгоритма. Не выполняется. Текст комментария ограничен символами {} или (*
*).



Азбука языка Pascal

- **Идентификатор** - имя для обозначения программ, а в программе переменных и постоянных величин, различных процедур, функций, объектов.
- **Константы** – данные, значения которых установлены в описательной части программы и в процессе выполнения не изменяются.
- **Переменные** – величины, которые могут менять свои значения в процессе выполнения программы.



Основные правила корректной записи программы на Турбо Паскаль:

- Избегайте запись текста сплошным стилем, т.е. при записи конструкций языка более глубоких уровней вложенности следует двигать их от начала строки вправо. Каждое описание и каждый оператор следует писать с новой строки.
- Любую программу сопровождать комментариями, поясняющими назначение всей программы и отдельных ее блоков, процедур, функций.
- Имена для объектов программы (переменные и константы) надо выбирать так, чтобы они соответствовали этим объектам.

Зарезервированные слова

Паскаль:

absolute	абсолютный
and	логический
array	массив
asm	ассемблер
begin	начало блока
case	вариант
const	константа
constructor	конструктор
div	деление нацело
go to	переход к
do	выполнять
downto	уменьшить
до	
destructor	разрушитель
else	иначе

end	и
exports	конец блока
external	внешний
file	файл
for	для
forward	
опережающий	
function	функция
if	если
implementation	
выполнение	
in	в
inline	
встроенный	
interrupt	прервать
interface	интерфейс
inherited	
унаследованный	

Паскаль:

label	метка
library	библиотека
mod	остаток от
деления	
nil	отсутствие
not	логическое НЕ
or	логическое
ИЛИ	
of	из
object	объект
(цель)	
packed	
упакованный	
procedure	процедура
program	
программа	
record	запись
repeat	
повторение	
set	
множество	

shl	сдвиг битов
влево	
shr	сдвиг битов
вправо	
string	строка
then	тогда
to	увеличивая
type	тип
unit	модуль
until	до
Uses	
использования	
var	переменная
while	пока
with	с
xor	исключающее
или	

Тема 4: Структура программы на языке Pascal

- **Исполнитель** – тот, кто выполняет приказы. **Формальный исполнитель** – не задумывается (компьютер), **Неформальный** – рассуждает (человек)
- **Команда** – указание исполнителю выполнить некоторое законченное действие.



Структура программы на языке Pascal

- **Каждая ли команда, записанная в программе выполняется?**
- нет, одни команды не выполняются, если имеется какое-либо условие на выполнение этого действия;
- другие команды всегда невыполнимы, но косвенно влияют на работу всей программы.
- Вся программа на языке TP разделена на модули по определенной схеме и каждый раздел имеет своё имя (**идентификатор**), которое в свою очередь дает команду или подсказку другой команде, являющейся исполнимой.

Структура программы на turbo

Pascal:

- **program** <имя программы>;
- **uses** <имя1 раздела, имя2 раздела>;
- **label** <имя1 метки, имя2 метки>;
- **const** <имя постоянной = значение>;
- **type** ...;
- **var** <имя переменной : тип>;
- **procedure** <имя процедуры>;
- <тело процедуры>;
- **function** <имя функции>;
- <тело функции>
- **begin**
- <операторы>
- **end.**

Структура программы на языке Pascal

- В разделе **program** присваивается имя программы (не файла), по которому эта программа может быть вызвана из другой программы.
- Разделы **begin ... end** обязательны. Это раздел является составным оператором и часто встречается в программе более одного раза.
- Раздел **var** резервирует место переменным в оперативной памяти. Их следует обязательно назвать и перечислить.
- Раздел **const** объявляет постоянные, хотя их величины есть в библиотеке программы
- Для подключения новых библиотек используется модуль **uses** (использования), где перечисляются имена библиотек, чьи команды в программе обязательно понадобятся.
- Команда **uses system** подключается автоматически.
- В конце программы обязательно ставится точка, после нее редактор ничего не видит в тексте.