|  |  |
| --- | --- |
| Absolute | Признак абсолютной переменной (расположенной по определенному адресу) |
| and | Логическая операция И |
| array | Начало типа-массив |
| asm | Начало блока на ассемблере |
| assembler | Подпрограмма, написанная на ассемблере |
| begin | Начало тела программы или подпрограммы, начало составного оператора |
| case | Начало условного оператора CASE |
| const | Идентификатор раздела констант, группы параметров-констант |
| constructor | Конструктор объекта |
| destructor | Деструктор объекта |
| div | Операция деления целых чисел с получением целого результата |
| do | Признак начала оператора в цикле FOR, WHILE, оператора WITH |
| downto | Указатель изменения параметра цикла от максимального к минимальному значению |
| else | Секция ИНАЧЕ в условном операторе IF, CASE |
| end | Конец тела программы, подпрограммы, конец модуля, конец составного оператора, блока на ассемблере, оператора CASE, типа-запись, объекта |
| external | Директива внешней подпрограммы |
| far | Директива формирования "дальнего" адреса |
| file | Файловый тип |
| for | Начало цикла с известным числом повторений |
| forward | Директива заголовка подпрограммы, описанной дальше по тексту программы |
| function | Начало подпрограммы-функции |
| goto | Безусловный переход |
| if | Начало условного оператора IF |
| implementation | Исполнительная часть модуля |
| in | Проверка наличия элемента в множестве |
| inherited | Наследуемый метод (только в версии 7.0) |
| inline | Директива, позволяющая включить в программу часть, написанную на ассемблере |
| interface | Интерфейс модуля |
| interrupt | Директива процедуры обработки прерываний |
| label | Идентификатор раздела меток |
| mod | Остаток от деления целых чисел |
| near | Директива формирования "ближнего" адреса |
| nil | Пустая ссылка |
| not | Логическое отрицание |
| object | Начало типа-объект |
| of | Начало типа компонентов в определении массива, множества, файла, начало перечисления констант условного оператора CASE |
| оr | Логическая операция ИЛИ |
| packed | Признак "упакованного" массива |
| private | Начало приватной секции объекта |
| procedure | Начало подпрограммы-процедуры |
| program | Начало программы |
| public | Начало доступной извне секции объекта (только в версии 7.0) |
| record | Начало типа-запись |
| repeat | Начало цикла REPEAT |
| set | Тип-множество |
| shl | Сдвиг битов влево с заполнением нулями освободившихся разрядов |
| shr | Сдвиг битов вправо с заполнением нулями освободившихся разрядов |
| string | Тип-строка |
| then | Раздел ТОГДА условного оператора IF |
| to | Указатель изменения параметра цикла от минимального к максимальному значению |
| type | Идентификатор раздела типов |
| unit | Начало модуля |
| until | Конец цикла REPEAT |
| uses | Указатель используемых модулей |
| var | Идентификатор раздела переменных, группы параметров-переменных |
| virtual | виртуальный метод |
| while | Начало цикла WHILE |
| with | Начало оператора над записями |
| xor | Логическая операция ИСКЛЮЧАЮЩЕЕ ИЛИ |

**Зарезервированые слова**

Функции

Abs(x) - Модуль аргумента

**Bar** - Рисует закрашенный прямоугольник, используя текущие стиль и цвет закраски.

**break** выполняет полный выход из цикла, т.е. все возможные итерации цикла прерываются.

Chr-Возвращает [символ](http://pascal.net.ru/Char) с определённым номером в [ASCII таблице](http://pascal.net.ru/866).

Circle - Рисует окружность текущим цветом, используя точку (X, Y) как центр и параметром Radius как радиус данной окружности.

ClearDevice - Очищает текущее устройство вывода и устанавливает текущий указатель в точку (0, 0).

CloseGraph - Закрывает графическую систему.

Continue - Продолжает циклы [for](http://pascal.net.ru/for), [while](http://pascal.net.ru/while) и [repeat](http://pascal.net.ru/repeat).

Copy(S , Index , Count) - возвращает подстроку строки S, содержащую Count символов, начиная с символа с номером Index.

Dec – уменьшение на 1.

Delete(S ,Index,Count)- удаляет Count символов из строки S, начиная с позиции Index.

Div – целочисленное деление.

Exit- Немедленно выходит из текущего блока программы (из [функции](http://pascal.net.ru/function) или [процедуры](http://pascal.net.ru/procedure)).

exp(x) - возвращает экспоненту параметра

FillEllipse(X, Y ,XRadius, YRadius) Рисует закрашенный эллипс.

Findfirst - Находит следующее вхождение файла, соответствующего имени и набору атрибутов, определенных в предшествующем обращении к процедуре FindFirst.

FindNext - Находит следующее вхождение файла, соответствующего имени и набору атрибутов, определенных в предшествующем обращении к процедуре FindFirst.

Frac - Возвращает дробную часть аргумента.

FreeMem - Освобождает память, занятую динамической переменной данного размера.

GetEnv - Возвращает значение заданной переменной среды DOS.

GetImage(X1, Y1, X2, Y2, BitMap)- Сохраняет участок изображения в буфер

GetMaxX - Возвращает текущее значение разрешения по горизонтали

GetMaxY - Возвращает текущее значение разрешения по вертикали.

GetMem(Var P : Pointer; Size : Word)-Создает динамическую переменную определенного размера и записывает ее адрес в памяти в указательную переменную.

GetPixel- Возвращает значение пиксела в точке с координатами (X, Y).

GraphResult - Возвращает код ошибки для последней графической операции.

Halt - Останавливает выполнение программы и выходит в операционную систему.

ImageSize - Возвращает число байт памяти, необходимых для сохранения заданной прямоугольной области экрана.

Inc – увеличивает аргумент на 1.

InitGraph - Инициализирует графическую систему и переводит видеокарту в графический режим

Insert(Source : String; Var S : String; Index : Integer)- вставляет строку Source в строку S в позиции с номером Index.

**Int** - Возвращает целую часть параметра.

Length - Возвращает динамическую длину строки.

Line(X1, Y1, X2, Y2) - Рисует линию из точки с координатами (X1, Y1) в точку с координатами (X2, Y2).

**LineRel** (x,y) рисует линию, у которой начальные координаты совпадают с координатами текущего указателя, а координаты конца высчитываются путем прибавления (x,y)

**Ln** - Возвращает натуральный логарифм аргумента.

Mod - вычисляет остаток от целочисленного деления

Move - Копирует байты из Source в Dest

Odd - Проверяет параметр на нечетность. Значение функции Odd(X) равно True, если X - нечетное число.

Ofs -Возвращает смещение определенного объекта.

Ord(X) - Возвращает порядковое значение выражения перечисляемого типа.

OutText - Посылает строку на устройство вывода в позиции текущего указателя.

Pos(Substr : String; S : String) -ищет первое вхождение строки Substr в строку S и возвращает целочисленное значение, которое является индексом первого символа Substr внутри S

Pred - Возвращает предшественника параметра.

Ptr - Преобразовывает адрес в виде сегмент:смещение в указатель

PutImage - Помещает битовое изображение на экран.

PutPixel- Ставит точку в позиции (X, Y).

Rectangle - Рисует прямоугольник, используя текущий тип и цвет линии.

**Round**: Округляет значение вещественного типа до значения целочисленного типа.

Seg - Возвращает сегмент определенного объекта

SetActivePage Устанавливает активную страницу для графического вывода.

SetColor Устанавливает текущий цвет, используя палитру. В режимах 16 и 256 цветов обычно можно использовать [константы цветов](http://pascal.net.ru/Black)

SetFillStyle - Устанавливает цвет и стиль закраски. В режимах 16 и 256 цветов обычно можно использовать [константы цветов](http://pascal.net.ru/Black).

Sizeof - определяет объем памяти, отводимого под элемент данных

Sqr – квадрат аргумента

Sqrt – квадратный корень аргумента

Str - Преобразовывает число в строку

Succ - Возвращает последователя параметра

TextColor - Выбирает цвет символов

**Trunc**: Усекает значение вещественного типа к значению целочисленного типа.

UpCase - Преобразовывает символ из нижнего в верхний регистр

Val - Преобразовывает строковое значение в его числовое представление

Работа с файлами

Текстовые файлы - text

Var in : text;

Assign(<переменная>, путь к файлу) – связывает внешний файл и переменной

Reset (имя файла) – открыть файл для чтения

Rewrite (имя файла) - открыть файл для записи

Close (имя файла) – закрыть файл

Read, readln(<имя файла>,спис. Перемен.) -считывает с файла и подключает к переменной

Write, writeln(<имя файла>,спис. Перемен.)- записывает в файл переменную

Append – открывает существующий файл связанный с файловой переменной для добавления в него новых записей

Chdir – выполняет смену текущего каталога на каталог маршрут к которову указан в текстовой переменной {$I+/-}

Erase(f) – стирает файл f

Eof – конец файла

Filepos – возвращает текущую позицию для файла связаного с файловой переменной. При положении текущего указателя в начале файла возвращает нулевое значение. **Для текстовых файлов не используется**

Filesize - Параметр F - файловая переменная. FileSize(F) возвращает число компонентов в F. Если файл пустой, то FileSize(F) возвращает нуль. **Для текстовых файлов не используется**

Flush - Если текстовый файл был открыт на вывод с использованием процедур ReWrite или Append, то вызов Flush очистит буфер файла. Это гарантирует, что все символы, записанные в это время в файл будут насильно записаны на диск. Вызов Flush не имеет никакого эффекта для файлов, открытых на ввод.

Getdir(D, f) -Возвращает текущий каталог заданного диска. Имя каталога находится в строковой переменной F, а диск задается значением параметра D: 0 - Текущий диск 1 - Диск A 2 - Диск B 3 - Диск C И так далее...

IOResul – возвращает целое значение являющееся состоянием последней выполненой операции ввода-вывода. Нулевое значение соответствует нормольному запершению операции

Mkdir - создает подкаталог имя которого задается переменной

Rename - переименовывает внешний файл связаный с файловой переменной присваивая ему имя содержащееся в строковой переменной

Rmdir – удаляет пустой каталог маршрут которого указан в строковой переменной

Seek – перемещает текущую позицию в файле связаном с файловой переменной на заданый элемент . началу файла соответствует нулевое значение . ДЛЯ ТЕКСТОВЫХ ФАЙЛОВ НЕ ИСПОЛУЗЕТСЯ.

Seekeof – возвращает состояние «конец файла» для текстового файла связаного с файловой переменной

Seekeoln – возвращает состояние конец строки для текстового файла связаного с файловой переменной

Settextbuf - назначает буфер ввода-вывода для текстового файла связаного с файловой переменной. Никогда не применяется к открытым файлам поскольку в этом случае возможны потери даных

Truncate - Удаляет часть файла, начиная с текущей позиции. Файл должен быть открыт процедурой Reset или Rewrite. Для настройки на компоненту файла, с которой следует удалить его часть, можно воспользоваться процедурой Seek. ДЛЯ ТЕКСТОВЫХ ФАЙЛОВ НЕ ИСПОЛЬЗУЕТСЯ.

Типизированые файлы

Var ftable: file of type\_id;

Type\_id – любой тип

Нетипизированые файлы

Var

Name: file;

**BlockRead** - Считывает одну или большее количество записей из файла в переменную.

**BlockWrite** - Записывает одну или большее количество записей из переменной в файл.

Записи

Type

Term = record

M1: real;

M2: array [1..3] of word;

End;

Var mr: term;

Begin

X:= mr.m1;

With mr do x:=m1;

End.

Type

Rec = record

V1,v2: integer;

Case n:word of

0: (ma: (1,2,3));

…

N: (pa: (x,r,t));

End;

Модуль

**Unit** <имя\_модуля>;

**interface  описание переменных костант функций процедур тд**;

**implementation**< исполняемая часть >;

**begin**

<инициирующая часть>;

**end** .

Типы

Integer -32768..32767

Byte 0..255

Word 0..65535

Shortint -128..127

Longint -2147483648..2147483647

Real 2,9x10-39 - 1,7x1038

Single 1,5x10-45 - 3,4x1038

Double 5x10-324 - 1,7x10308

Extended 3,4x10-4932 - 1,1x104932

Boolean true, false

Char все символы кода ASCII

String строковый тип [255]

Операторы

If .. then..else

Case .. of

End;

While .. do

Repeat

Исполняемая часть

Until контролер;

For .. to .. do

For .. downto .. do

Uses модули

**CRT**

delay - Приостанавливает выполнение программы на заданное число миллисекунд (пауза, задержка).

Clrscr- Очищает активное окно и устанавливает курсор в левый верхний угол окна.

Clreol - Очищает все символы в строке с позиции курсора до конца строки без перемещения курсора.

gotoXY - Перемещает курсор в заданную позицию экрана.

Keypressed - Определяет, была ли нажата клавиша на клавиатуре.

Readkey - Читает символ из буфера клавиатуры. Символ не отображается на экране.

Nosound - Выключает внутренний динамик компьютера.

Sound - Включает внутренний динамик.

Textcolor - Выбирает цвет символов

Wherey - Возвращает текущую координату Y курсора

**DOS**

GetDate Возвращает текущую дату, установленную в операционной системе.

[GetFTime   (процедура)](http://pascal.net.ru/GetFTime)  Возвращает дату и время обновления файла.  
 [GetTime    (процедура)](http://pascal.net.ru/GetTime)  Возвращает текущее время в операционной системе.  
 [PackTime   (процедура)](http://pascal.net.ru/PackTime)  Преобразовывает запись типа DateTime в LongInt.  
 [SetDate    (процедура)](http://pascal.net.ru/SetDate)  Устанавливает текущую дату в операционной системе.  
 [SetFTime   (процедура)](http://pascal.net.ru/SetFTime)  Устанавливает атрибуты даты и времени файла.  
 [SetTime    (процедура)](http://pascal.net.ru/SetTime)  Устанавливает текущее время в операционной системе.  
 [UnpackTime (процедура)](http://pascal.net.ru/UnpackTime)  Преобразовывает 4-х байтовое, упакованное значение  
                         типа Longint, полученное от [GetFTime](http://pascal.net.ru/GetFTime), [FindFirst](http://pascal.net.ru/FindFirst),  
                         или [FindNext](http://pascal.net.ru/FindNext) в запись типа DateTime.

[DiskFree (функция)](http://pascal.net.ru/DiskFree)  Возвращает число свободных байт на диске.  
 [DiskSize (функция)](http://pascal.net.ru/DiskSize)  Возвращает полный размер диска.

[FExpand   (функция)](http://pascal.net.ru/FExpand)   Расширяет имя файла до полного.  
 [FSearch   (функция)](http://pascal.net.ru/FSearch)   Ищет файл.  
 [FSplit    (функция)](http://pascal.net.ru/FSplit)   Разбивает имя файла на три его компонента.  
 [FindFirst (процедура)](http://pascal.net.ru/FindFirst)   Ищет файл в заданном каталоге.  
 [FindNext  (процедура)](http://pascal.net.ru/FindNext)   Ищет следующий файл, соответствующий имени  
                         и атрибутам, найденного ранее файла.  
 [GetFAttr  (процедура)](http://pascal.net.ru/GetFAttr)   Возвращает атрибуты файла.  
 [SetFAttr  (процедура)](http://pascal.net.ru/SetFAttr)   Устанавливает атрибуты файла.

[GetIntVec (процедура)](http://pascal.net.ru/GetIntVec)   Возвращает адрес, сохраненный в заданном  
                         векторе прерывания.  
 [Intr      (процедура)](http://pascal.net.ru/Intr)   Выполняет определенное программное прерывание.  
 [MsDos     (процедура)](http://pascal.net.ru/MsDos+(2))   Выполняет вызов функции DOS.  
 [SetIntVec (процедура)](http://pascal.net.ru/SetIntVec)   Устанавливает заданный вектор прерывания  
                         на заданный адрес.

[Exec        (процедура)](http://pascal.net.ru/Exec)   Выполняет заданную программу с заданными  
                           параметрами командной строки.  
 [Keep        (процедура)](http://pascal.net.ru/Keep)   Завершает программу и оставляет ее в памяти.  
 [SwapVectors (процедура)](http://pascal.net.ru/SwapVectors)   Переключает указатели SaveIntXX в модуле  
                           System на текущие вектора.  
 [DosExitCode (функция)](http://pascal.net.ru/DosExitCode)   Возвращает код завершения подпроцесса.

Множества

Var s: set of <type> or <diapos>

Const s=[1..8]

If n in( належить ) s then true else false

Include(s, ch); - s:= s+[ch];

Exclude(s, ch); - s:= s – [ch];

Директивы

Директивы компилятора

 - Начинаются со знаков "{$" или "(\*$"  
 - Сопровождаются именем директивы  
 - Заканчиваются знаками "}" или "\*)"

Они могут быть локальными или глобальными.

 - Локальные директивы могут находиться где-нибудь в теле программы.  
   Они влияют только на часть компиляции.  
 - Глобальные дирктивы должны находиться перед разделом описаний  
   программы или модуля. Они влияют на всю компиляцию.

[$A](http://pascal.net.ru/%24A)  Байтовое выравнивание данных - переключает между байтовым выравниванием и выравниванием по словам переменных и типизированных констант (Глобальная)

[$B](http://pascal.net.ru/%24B)  Булева оценка - Переключает между двумя различными моделями генерации объектного кода для булевых операторов **AND** и **OR**. (локальная)  
 [$D](http://pascal.net.ru/%24D)  Включает или выключает генерацию отладочной информации. (глобальная)

[$E](http://pascal.net.ru/%24E)  Эмуляция 80x87 - Включает или выключает компоновку библиотеки программной эмуляции 80x87 в EXE файл.(глобальная)

[$F](http://pascal.net.ru/%24F)  Дальний тип вызова - Задает модель вызова, используемую для последовательно компилируемых процедур и функций.(локальная)

[$G](http://pascal.net.ru/%24G+%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F+%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D0%B8%D0%B9+80x286)  Группа модулей в сегменте Определяет группу модулей, которые компоновщик должен поместить в один и тот же сегмент.(локальная)  
 [$I](http://pascal.net.ru/%24I)  Проверка ввода/вывода - Включает или выключает автоматическую генерацию объектного кода, которая проверяет результат обращения к процедуре ввода/вывода.(локальная)  
 [$L](http://pascal.net.ru/%24L)  Информация о локальных символах - Включает или выключает генерацию информации о локальных символах.(глобальная)

[$N](http://pascal.net.ru/%24N)  Математический сопроцессор - Переключает между двумя различными моделями генерации объектного кода обработки чисел с плавающей точкой, обеспечиваемыми компилятором.(глобальная)

[$O](http://pascal.net.ru/%24O)   генерация оверлейного кода - Включает и выключает генерацию оверлейного объектного кода.(глобальная)  
 [$P](http://pascal.net.ru/%24P)   Открытые строковые параметры - Управляет значением переменных параметров, объявленных с использованием зарезервированного слова **String(глобальная)**

[$Q](http://pascal.net.ru/%24Q)  Проверка переполнения - Управляет генерацией кода проверки переполнения.(локальная)

[$R](http://pascal.net.ru/%24R)  Проверка диапазона - Включает и выключает генерацию кода проверки диапазона.(локальная)

[$S](http://pascal.net.ru/%24S)  Проверка переполнения стека - Включает и выключает генерацию кода проверки переполнения стека.(локальная)  
 [$T](http://pascal.net.ru/%24T)  Проверка типов указателей - Управляет типами указателей, созданных [**оператором @.**](http://www.pascal.helpov.net/index/pascal_@)(локальная)

[$V](http://pascal.net.ru/%24V)  Проверка строковых параметров - Управляет проверкой типов строк, переданных как переменные параметры.(Локальная)

[$W](http://pascal.net.ru/%24W)  Область стека Windows - Генерирует специальные пролог- и эпилог- коды для дальних процедур и функций.(локальная)

[$X](http://pascal.net.ru/%24X)  Расширенный синтаксис - Включает или выключает расширенный синтаксис Turbo Pascal(глобальная)

[$Y](http://pascal.net.ru/%24Y)  Информация о символах - Включает или выключает генерацию информации о ссылках на символы.(глобальная)

Вы можете группировать несколько переключателей, разделяя их запятыми. Например:

 {$F+, $R+, $E-, $D-}

Масивы

<name> : array [1..3,1..5] of <type>;

<name>[k]

<name>[k,j]

Указатель

Var

Pn : ^integer;

Begin

Pn:=@n; присвоили адрес n для pn

Pn^ то на что указывает pn

New(pn); создаёт пам\*ять под переменную

Dispose(pn); удалить выделеную пам\*ять

Pn:=nil; присваетвает ничего

End.