

## Язык Паскаль

Язык программирования *Паскаль*, названный в честь французского математика и философа Блеза Паскаля (1623-1662), был создан как учебный язык программирования в 1968-1971 годах *Никлаусом Виртом* в Высшей технической школе (ETH) в Цюрихе.

Основные принципы Паскаля таковы:

*Структурное программирование.* Суть его заключается в оформлении последовательностей команд как замкнутых функций или процедур и в объединении данных, связанных по смыслу, в сложные структуры данных. Благодаря этому повышается наглядность текста и упрощается его отладка.

*Возможность проектирования сверху вниз.* Программист разбивает свою задачу на несколько более простых, после чего каждая из задач решается по отдельности. Затем komponуются результаты проектирования простых задач и решается задача проектирования сверху вниз.

Горячая клавиша	Эквивалентная команда меню	Функция
F1	—	Активизация окна контекстно-зависимой помощи
Alt + F1	Help / Previous topic	Возврат к предыдущей справке
Ctrl + F1	Help / Topic search	Активизация синтаксической справки, т.е. справки об операторе, на который указывает маркер
Shift + F1	Help / Index	Вызов содержания справочной системы
F2	File / Save	Сохранение на диске файла из активного окна
Ctrl + F2	Run / Program Reset	Установка программного счетчика на начало программы и закрытие всех ранее открытых программой файлов
F3	File / Open	Открытие нового окна и загрузка в него выбранного файла
Alt + F3	Window / Close	Закрытие активного окна
Ctrl + F3	Debug / Call Stack	Открытие окна протокола используемых процедур
F4	Run / Goto Cursor	Выполнение программы, расположенной в активном окне, до позиции курсора
Ctrl + F4	Debug / Evaluate / Modify	Просмотр и изменение значений переменных
F5	Window / Zoom	Изменение размера активного окна
Alt + F5	Debug / User Screen	Переключение на пользовательский экран
Ctrl + F5	Window / Size / Move	Изменение положения и размера окна
F6	Window / Next	Переход к следующему окну
Shift + F6	Window / Previous	Возврат к предыдущему окну
F7	Run / Trace Into	Трассировка программы пооператорно с пооператорным выполнением всех подпрограмм
Alt + F7	Tools / Go to previous	Переход к предыдущей строке в окне сообщений
Ctrl + F7	Debug / Add watch	Дополнение списка переменных в окне Watch

F8	Run / Step over	Трассировка программы пооператорно с выполнением подпрограмм без пооператорной детализации
Alt + F8	Tools / Go to next	Переход к следующей строке в окне сообщений
Ctrl + F8	—	Установка / отмена контрольной точки на строке программы, указываемой курсором
F9	Compile / Make	Компиляция и редактирование связей программы
Alt + F9	Compile / Compile	Компиляция программы из активного окна
Ctrl + F9	Run / Run	Компиляция и выполнение программы
F10	—	Активизация строки меню
Alt + F10	—	Вызов локального меню
Alt + литера	—	Открытие озаглавленного выбранной литерой подчиненного меню из строки меню
Alt + Backspace	Edit / Undo	Отмена всех изменений в текущей строке
Alt + X	File / Exit	Завершение сеанса работы
Alt + цифра	—	Переход к окну с указанным номером
Alt + 0	Window / List	Вызов окна, в котором содержится список всех открытых окон
Ctrl + Del	Edit / Clear	Удаление выделенного блока
Ctrl + Ins	Edit / Copy	Копирование выделенного блока в буфер промежуточного хранения
Shift + Del	Edit / Cut	Перенос выделенного блока в буфер промежуточного хранения
Shift + Ins	Edit / Paste	Копирование блока из буфера промежуточного хранения в окно редактирования

## Операции

Символ операции	Название операции	Тип допустимых операндов	Тип результата
<i>Операторы, имеющие первый приоритет</i>			
@	Взятие адреса	Любой	pointer
-	Унарный минус	Целый, вещественный	Целый, вещественный
+	Унарный плюс	Целый, вещественный	Целый, вещественный
NOT	Поразрядное "НЕ"	Целый	Целый
NOT	Логическое "НЕ"	Логический	Логический
<i>Операторы, имеющие второй приоритет</i>			
*	Умножение	Целый * Целый	Целый
		Целый * Вещественный	Вещественный

		Вещественный * Целый	Вещественный
		Вещественный * Вещественный	Вещественный
/	Деление	Целый / Целый	Вещественный
		Целый / Вещественный	Вещественный
		Вещественный / Целый	Вещественный
		Вещественный / Вещественный	Вещественный
div	Целочисленное деление	Целый div Целый	Целый
mod	Остаток от деления	Целый mod Целый	Целый
and	Логическое "И"	Логический and Логический	Логический
and	Поразрядное "И"	Целый and Целый	Целый
shr	Циклический сдвиг вправо	Целый shr Целый	Целый
shl	Циклический сдвиг влево	Целый shl Целый	Целый
<i>Операторы, имеющие третий приоритет</i>			
+	Сложение	Целый + Целый	Целый
		Целый + Вещественный	Вещественный
		Вещественный + Целый	Вещественный
		Вещественный + Вещественный	Вещественный
-	Вычитание	Целый - Целый	Целый
		Целый - Вещественный	Вещественный
		Вещественный - Целый	Вещественный
		Вещественный - Вещественный	Вещественный
or	Логическое "ИЛИ"	Логический or Логический	Логический
or	Поразрядное "ИЛИ"	Целый or Целый	Целый
xor	Логическое исключающее "ИЛИ"	Логический xor Логический	Логический
xor	Поразрядное исключающее "ИЛИ"	Целый xor Целый	Целый
<i>Операторы, имеющие четвертый приоритет</i>			
in	Вхождение во множество	Тип_элемента_множества in Множество	Логический
>	Больше		Логический
<	Меньше		Логический
=	Равно		Логический

<>	Не равно		Логический
>=	Больше или равно		Логический
<=	Меньше или равно		Логический

### Математические функции

Функция	Тип аргумента	Тип результата	Действие
Abs(x)	Целый, вещественный	Целый, вещественный	Возвращает абсолютное значение числа x
Cos(x)	Целый, вещественный	Вещественный	Возвращает косинус числа x, где x - угол в радианах
Sin(x)	Целый, вещественный	Вещественный	Возвращает синус числа x, где x - угол в радианах
Arctan(x)	Целый, вещественный	Вещественный	Возвращает арктангенс числа x
Exp(x)	Целый, вещественный	Вещественный	Возвращает число, равное $e^x$
Ln(x)	Целый, вещественный	Вещественный	Возвращает число, равное $\ln x$
Pi	-	Вещественный	Число $\pi$
Sqr(x)	Целый, вещественный	Вещественный	Возвращает число $x^2$
Sqrt(x)	Целый, вещественный	Вещественный	Возвращает число $\sqrt{x}$
Trunc(x)	Вещественный	LongInt	Возвращает число, равное целой части числа x. Округление происходит путем отбрасывания дробной части числа x
Frac(x)	Вещественный	Вещественный	Возвращает число, равное дробной части числа x
Int(x)	Вещественный	Вещественный	Возвращает число, равное целой части числа x
Round(x)	Вещественный	LongInt	Возвращает число, равное целой части числа x. Округление происходит по правилам математики, то есть к ближайшему целому
Random(x)	Целый	Вещественный	Возвращает случайное число от 0 до x. Эта функция может быть задана без аргумента Random. В этом случае будут генерироваться случайные числа от 0 до 1
Odd(x)	Целый	Логический	Возвращает значение TRUE (истина), если число x нечетно

## Математические процедуры

Inc(X,Y) - увеличивает значение числа X на Y. Процедура Inc(X) (без аргумента Y) увеличивает значение числа X на 1.

Dec (X,Y) - уменьшает значение числа X на Y. Процедура Dec(X) (без аргумента Y) уменьшает значение числа X на 1.

Randomize - обеспечивает несовпадение последовательностей случайных чисел, генерируемых функцией Random.

**Числа** записываются в десятичной системе счисления и могут быть целыми и вещественными (действительными), положительными и отрицательными. **Действительные числа** изображаются в естественной и показательной формах. Дробная часть отделяется от целой точкой.

Естественная форма : 4 .7 -0.71 +151.75 15.00 0.0067

**Показательная форма** записи используется для представления очень больших или очень маленьких чисел, которые нелегко изобразить обычной десятичной записью. При записи таких чисел часть числа, которая читается как *умножить на десять в степени*, обозначается буквой E, и число в программе выглядит следующим образом:

-0.131E5 (-0.131 · 10<sup>5</sup> = -13 100)

50E7 (50 · 10<sup>7</sup> = 500 000 000)

-5E-02 (-5 · 10<sup>-2</sup> = -0.05)

123.7E-5 (123.7 · 10<sup>-5</sup> = 0.001237)

**Строка** — это последовательность символов языка, заключенная в апострофы (одиночные кавычки): 'A', '123456'

**Имя (идентификатор)** служит для обозначения различных объектов и конструкций программы. Имя должно состоять из латинских букв и цифр, но первым символом может быть только буква: A, B1, Gamma

Имя может быть служебным, стандартным или определяемым пользователем.

**Служебные слова** используются для обозначения операторов, описаний некоторых операций. Назначение служебных слов строго определено, и они не могут быть использованы с другой целью.

Назначение **стандартных имен** также заранее определено, однако программист может переопределить эти имена, но делать это не рекомендуется. Стандартные имена используются при определении стандартных констант, типов переменных, процедур и функций.

**Имена, определяемые пользователем**, назначаются самим программистом и используются для обозначения данных — величин, над которыми производятся некоторые действия. Постоянные величины, значения которых не меняются в процессе выполнения программы, называются **константами**. Величины, значения которых меняются в процессе выполнения программы, называются **переменными**.

## ОБЩАЯ СТРУКТУРА ПРОГРАММЫ

**PROGRAM** ИМЯ ПРОГРАММЫ ;

  LABEL МЕТКА 1; ...;МЕТКА N;

  CONST ИМЯ КОНСТАНТЫ = ЗНАЧЕНИЕ;

  ...;

  TYPE ИМЯ ТИПА= ТИП;

  ...;

  VAR ИМЯ ПЕРЕМЕННОЙ1, ..., ИМЯ ПЕРЕМЕННОЙ N: ТИП ;

  ...;

  БЛОК ОПИСАНИЙ ПРОЦЕДУР;

  БЛОК ОПИСАНИЯ ФУНКЦИЙ;

**BEGIN**

  ТЕЛО ПРОГРАММЫ

**END.**

## Операторы работы с файловым типом данных

В программе на Паскале текстовый файл представлен файловой переменной типа TEXT.

  Var <имя файловой переменной>: text;

Связь файловой переменной с именем файла на диске осуществляется с помощью оператора

Assign (имя файловой переменной, имя файла на диске);

Имя файла на диске больше нигде в программе не появляется. <Имя файла на диске> должно содержать полный путь для поиска данного файла. Если полный путь к файлу не указан, то файл будет искаться в каталоге, где находится исходный текст программы.

Открытие файла для чтения выполняется оператором

Reset (имя файловой переменной);

Если на диске нет файла с данным именем, то выполнение программы прервется.

Открытие файла для записи осуществляется оператором

Rewrite (имя файловой переменной);

Если на диске нет файла с данным именем, то файл будет создан.

Открытие файла для дозаписи осуществляется оператором

Append (имя файловой переменной);

Если на диске нет файла с данным именем, то выполнение программы прервется.

Окончание работы с файлом фиксируется с помощью оператора

Close (имя файловой переменной);

### Пример программы

```
var
  a, b, c: integer;
  f, g: text;
begin
  assign (f, 'input.txt');
  reset (f);
  readln (f, a, b);
  close (f);
  c:=a+b;
  assign (g, 'output.txt');
  rewrite (g);
  writeln (g, c);
  close (g);
end.
```

```
{Описываем файловые переменные f и g.}
{Устанавливаем связь программы с файлом на диске.}
{Открываем файл для чтения.}
{Читаем две переменные целого типа из файла.}
{Закрываем файл.}
{Устанавливаем связь программы с файлом на диске.}
{Открываем файл для записи.}
{Записываем значение переменной c в файл.}
{Закрываем файл.}
```

### Функции преобразования

Функция	Описание
Chr	Возвращает символ, заданный целым числом.
High	Возвращает старшее значение в диапазоне аргумента.
Low	Возвращает младшее значение в диапазоне аргумента.
Ord	Возвращает порядковое число по значению перечислимого типа.
Round	Округляет значение вещественного типа до значения, имеющего длинный целый тип.
Trunc	Усекает значение вещественного типа до значения, имеющего длинный целый тип.

Следующие процедуры и функции используются для работы со строками Паскаля.

Процедура	Описание
Concat	Выполняет конкатенацию последовательности строк.
Copy	Возвращает подстроку строки.
Delete	Удаляет из строки подстроку.
Insert	Добавляет в строку подстроку.
Length	Возвращает динамическую длину строки.
Pos	Производит поиск подстроки в строке.
Str	Преобразует численное значение в его строковое представление.
Val	Преобразует строковое значение в его численное представление.

#### Использование модуля CRT

Чтобы использовать модуль Crt, его нужно указать в операторе uses:

```
uses Crt;
```

Все координаты экрана, кроме тех, которые используются для определения окна, относятся к текущему окну. Координата экрана (1,1) соответствует левому верхнему углу экрана.

По умолчанию окном считается весь экран.

#### Специальные символы

При записи в выходной файл или в файл, который назначен для модуля Crt, специальное значение имеют следующие управляющие символы:

Символ	Название	Описание
#7	Звонок BELL	Вызывает звуковой сигнал, издаваемый с помощью внутреннего динамика.
#8	Обратный пробел BS	Возврат на одну позицию. Вызывает перемещение курсора влево на одну позицию. Если курсор уже находится у левого края текущего окна, то никаких действий не производится.
#10	Перевод строки LF	Перемещает курсор на одну строку вниз. Если курсор уже находится на нижней строке окна, то окно пролистывается вверх на одну строку.
#13	Возврат каретки BS	Возвращает курсор с левому краю текущего окна.

Процедуры и функции модуля Crt

Функция/процедура	Описание
AssignCrt	Назначает текстовый файл для устройства CRT.
ClrEol	Очищает все символы, начиная от позиции курсора до конца строки, без перемещения курсора.
ClrScr	Очищает экран и помещает курсор в верхнем левом углу.
Delay	Выполняет задержку на указанное число миллисекунд.
DelLine	Удаляет строку, на которой находится курсор и перемещает все следующие строки на одну строку вверх. Нижняя строка очищается.
GotoXY	Выполняет позиционирование курсора. X - это горизонтальная позиция, Y - вертикальная позиция.
HightVideo	Выбирает символы с подсветкой.
InsLine	Вставляет пустую строку в месте расположения курсора.
KeyPressed	Возвращает значение True, если клавиша на клавиатуре нажата и False - в противном случае.
LowVide	Выбирает символы с пониженной яркостью.
NormVideo	Выбирает символы с нормальной яркостью.
NoSound	Выключает внутренний динамик.
Sound	Включает внутренний динамик.
TextBackground	Выбирает фоновый цвет.
TextColor	Выбирает цвет самого символа.
TextMode	Выбирает конкретный текстовый режим.
Window	Определяет на экране текстовое окно.
Readkey	Считывает символ с клавиатуры.
WhereX	Возвращает координату X для текущей позиции курсора, относящуюся к текущему окну. X представляет собой горизонтальную позицию.
WhereY	Возвращает координату Y для текущей позиции курсора, относящуюся к текущему окну. Y представляет собой вертикальную позицию.