Object Pascal

Алфавит Паскаля

- -Все прописные и строчные буквы латинского алфавита;
- -Цифры 0..9;
- -Знаки арифметических операций * / + -;
- -Знаки операций сравнения: = <>>=><=<
- -Разделители: () [] {}.,;:^':=_;
- Специальные символы.

Выражения. Оператор присваивания

```
Общий вид выражения:
```

```
<oперанд1> <знак операции>
<oперанд2>
```

Чтобы переменной придать значение, используют оператор присваивания:

```
Inpumep:
```

```
D1:=SIN(t)-2*t;
I:=I+1;
```

Математические операции

Обозначение	Операция	Типы операндов	Примеры
+	Сложение	Целые, вещественные	A := A + B;
-	Вычитание	Целые, вещественные	C := B - 7;
*	Умножение	Целые, вещественные	A := A*1.5;
	Деление	Вещественные	H := H/2;
DIV	Целочисленное деление	Целые	I:=7 div 2;
MOD	Остаток от целочисленного деления	Целые	k:=7 mod 3;

Стандартные функции (system)

Мат. обозначение	Обозначение в программе	Название функции	Пример
$ \mathbf{X} $	abs(x)	Модуль	A:=abs(x);
X^2	Sqr(x)	Квадрат	B:=sqr(x);
$\sqrt{\mathbf{X}}$	Sqrt(x)	Корень	C:=sqrt(x);
sin x	Sin(x)	Триго-	$A1:=\sin(x);$
cos x	Cos(x)	номет-	
arctg x	Arctan(x)	рические	
e ^x	Exp(x)	экспонента	$D:=\exp(x);$
ln x	Ln(x)	Натуральный логарифм	C:=ln(x);
	Round(x)	Округление	I:=round(c);
	Trunc(x)	Отсечение	i:=trunc(x);

Комментарий

Комментарии используются для пояснения фрагментов программы.

Комментарий - это последовательность любых символов, заключенная между фигурными скобками.

```
{ Это комментарий }
{ А это пример многострочного 
Комментария
}
```

Комментарий можно использовать в конце строки.

Идентификаторы

 Идентификатор (имя) служит для обозначения программных объектов: констант, типов, переменных, процедур, функций. Имя является ссылкой на используемый в программе объект. Имя состоит из букв, цифр и может включать символ подчеркивания " ", начинается обязательно с буквы. Имена не должны совпадать со служебными словами.

- Переменная это именованный программный объект, который может изменять свое значение в ходе выполнения программы. Имя переменной является носителем ее значения.
- Константа это программный объект, который не изменяет своего значения в ходе выполнения программы. В Паскале допускаются неименованные и именованные константы, т.е. константа в программе может обозначаться именем, которому ставится в соответствие неизменяемое значение.

Заголовки разделов

Разделы описаний в модуле появляются только в том случае, если в программе необходимы программные объекты, которые в этих разделах описываются: **USES** – Список подключенных модулей; LABEL - Метки; **CONST** – Список констант; **ТҮРЕ** – Описание типов; **VAR** – Описание переменных; **FUNCTION, PROCEDURE;**

ОПИСАНИЕ ДАННЫХ все программные объекты должны быть описаны до их первого использования.

ЦЕЛЫЙ ТИП INTEGER – имя типа;

Операции

Обозначения

DIV

- Сложение
 - Вычитание
- Умножение
- Деление нацело
- Взятие остатка от деления MOD

Примеры:

Запись	Результат	
7 div 2	3	
3 div 5	0	
7 mod 2	1	

СТАНДАРТНЫЕ ФУНКЦИИ

 Обозначение
 Тип аргумента
 Результат

 ABS(x) —
 целый
 модуль х

 SQR(x) —
 целый
 квадрат х

 TRUNC(x) —
 вещественный
 целая часть х

 ROUND(x) —
 вещественный
 округление до целого

• <u>Константы</u>: обычная математическая запись целых чисел, например: 3, 276, -19.

Вещественный тип

Имя типа: REAL

- Допустимые операции: сложение, вычитание, умножение, деление
 - Стандартные функции:

Abs(x), cos(x), sin(x), sqr(x), exp(x), arctan(x), ln(x), sqrt(x)

<u>Константы</u>: используются две формы записи:

- с фиксированной точкой (обычная математическая запись, но дробная часть от целой отделяется точкой);
- с плавающей точкой (вещественное число представляется в виде числового коэффициента, умноженного на степень по основанию 10, при записи вместо основания степени 10 используется буква е или Е).
- Примеры: 2.87, 163.55, 0.93, 0.287E 01, 287e-02, 43e-23.

Логический тип

Имя типа: BOOLEAN Логические операции:

- NOT логическое отрицание;
 - AND логическое И;
 - OR логическое ИЛИ;
 - Операции отношения:

 Логические операции дают в результате логическое значение, которое можно определить по значениям данных из

табл.:

а	ь	a AND b	a OR b
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Операции отношения для любых типов данных также дают в результате логическое значение.

Константы: False, True

Литерный (символьный) тип

Имя типа: CHAR

• Операции отношения:

• Стандартные функции:

ORD(c) – порядковый номер (код) литеры;

СНR(i) – литера с номером i.

<u>Константы</u>: литера в апострофах, например '9', '+', 'A'

- Наряду со стандартными типами данных, в языках программирования вводится и новые, нестандартные простые типы данных, множество значений которых задается перечислением этих значений. Для ввода в употребление любых новых типов в программе существует раздел описания типов, для указания которого в языке используют служебное слово "type", а описание новых типов дается в следующей форме:
- <имя типа> = <определение типа>;

Перечислимый тип

описание перечислимых типов задается в виде: <имя типа> = (<перечисление значений>).

<u>Имя типа</u>: задается программистом, вводится в употребление в разделе описания типов.

Пример:

type

```
дни_недели = (пн, вт, ср, чтв, пят, суб, вск);

звукоряд = (до, ре, ми, фа, соль, ля, си);

фрукты = (яблоко, груша, персик, айва,

апельсин);
```

Ограниченный (диапазонный) тип

- В Паскале ограниченный тип задается на основе ранее описанного перечислимого или стандартного типа указанием диапазона. Диапазон указывается двумя константами базового типа (нижняя и верхняя границы), разделенных двумя точками.
- Пример:
- type
- рабочие_дни = пн..пят;
- индекс = 1..20;
- var
- день_работы: рабочие_дни;
- день отдыха : суб..вск;

Регулярные типы (массивы)

Тип массива имеет вид: ARRAY [тип индекса] ОF [тип элементов]; Одномерный массив:

VAR

Temperatura: ARRAY [1..40] OF REAL; Двумерный массив:

ARRAY [тип индекса 1, тип индекса 2] OF [тип элемента];

Пример двумерного массива

type

```
погода = ( ясно, пасмурно, дождь, снег);
месяц = (янв, фев, март, апр, май, июнь, июль,
авг, сент, окт, нояб, дек);
день = 1..31;
```

var

погода_дня : array [день, месяц] of погода;

Над массивами определена единственная операция операция присваивания, т.е. значение массива можно присвоить другому массиву того же типа.

Строковый тип

Короткая строка SHORTSTRING или STRING[n], n <= 255;

Длинная строка STRING

ограничена только имеющейся оперативной памятью.

Каждая строка трактуется как одномерный массив символов, но количество символов может быть разным. Так строки типа **String**[N] может меняться от 0 до N символов (ShortString – от 0 до 255 символов).

Комбинированные типы (записи)

■ Записи (комбинированные типы данных) совокупность разнородных, в общем случае, данных. Такая совокупность является единым программным объектом и имеет единое имя. Записи используются для представления в программе сложных объектов реального мира, обладающих совокупностью разнородных характеристик. Элементы записи называются полями записи.

В Паскале задание записи имеет вид:

record

```
<имя_поля_1>: <тип_поля_1>;
<имя_поля_2>: <тип_поля_2>;
<имя_поля_N>: <тип_поля_N>
end;
```

Каждое поле имеет свое уникальное имя и произвольный тип, в том числе им может быть и тип записи (такие записи называются иерархическими).

Пример описания записи, содержащей анкетные сведения о человеке:

```
type
  анкета = record
  \PhiAM : string;
  год_рожд: integer;
  адрес: record
  улица: string;
  номер_дома: integer
end
end;
 На основе введенного типа могут быть описаны переменные,
которые можно использовать для обработки разнородных
данных:
var
  карта_1, карта_2: анкета;
```

- Чтобы получить доступ к соответствующему полю переменной типа запись, нужно записать селектор записи, который имеет вид:
- < имя_переменной >.<имя_поля>
- Тогда справедливы операторы, использующие следующие обращения к полям записи:
- карта_1.год_рожд := 1973 ;
- карта_1.адрес.улица := 'Кузнечный взвоз'

Конструкции языка для описаний

1. Описание констант

Const

P = 3.14;

k = 1;

2. Описание переменных

Var

a,b,c: Real;

I,J: Integer;

Ввод/вывод данных

Ввод данных

Read – ожидание ввода с клавиатуры.

ReadIn — —//— с последующим переводом курсора на следующую строку.

Readkey – ожидание нажатия любой клавиши.

Read(x,y)

Вывод данных

- Write вывод на экран.
- WriteIn –// с переводом курсора на следующую строку.
- Пример
- writeln('x=',x:3:3,' y=',y:3:3);
- write('X=',X);