

CAPITULO 2.- TIPOS DE DATOS

LA SECCION DE DECLARACION DE UNIDADES

La versión 5.0 de Turbo Pascal tiene ocho unidades predefinidas, llamadas **UNIDADES ESTANDAR**. Estas unidades contienen procedimientos y funciones propias de Turbo Pascal.

Los nombres de estas unidades son: SYSTEM, DOS, OVERLAY, CRT, PRINTER, GRAPH, TURBO3 y GRAPH3.

La primera de ellas, la unidad SYSTEM, contiene las declaraciones y definiciones esenciales que todos los programas necesitan. Turbo Pascal encadena automáticamente los recursos que esta unidad provee cuando se compila un programa. La unidad SYSTEM no debe ser incluida explícitamente por medio del estatuto USES; esto genera un error.

El resto de las unidades estándar contienen rutinas que pueden seleccionarse en cada programa. Para usar cualquiera de estas unidades, se debe incluir el nombre de la unidad en la zona del estatuto USES. A continuación se da una breve descripción del contenido de estas unidades:

DOS. - Contiene las rutinas que ayudan a elaborar programas asociados con el sistema operativo de disco. Por ejemplo: GETDATE y GETTIME, procedimientos para acceder la fecha y la hora del sistema.

OVERLAY. - Pequeño grupo de rutinas requeridas para la implementación de código traslapable en Turbo Pascal.

CRT. - Incluye una variedad de rutinas comunes para controlar el video y el teclado. Por ejemplo: CLRSCR, CLREOL, GOTOXY, etc.

PRINTER. - Contiene la técnica para enviar datos a la impresora, en lugar del video. PRINTER define un archivo de texto con el nombre de LST. Cuando se utiliza WRITE o WRITELN direccionado a este archivo, Turbo Pascal lo redirecciona a la impresora.

GRAPH. - Es una colección grande de rutinas de graficación.

TURBO3 y GRAPH3. - Contienen las rutinas de la versión 3.0 de Turbo Pascal. En la mayoría de los casos, las rutinas de estas unidades han sido reemplazadas por procedimientos nuevos en la versión 5.0; sin embargo, se incluyen éstas para efectos de compatibilidad de código con la versión 3.0.

Por conveniencia, cinco de estas ocho unidades están "empacadas" en un archivo de librería llamado TURBO.TPL. Estas unidades son: SYSTEM, DOS, OVERLAY, CRT y PRINTER.

CONSTANTES

Las constantes son valores que se "incluyen" en el código fuente y no cambian durante la ejecución del programa. Hay dos tipos de constantes en el Pascal estándar: literales y declaradas. TURBO Pascal provee un tipo más de constante, aunque en realidad no es tal. Se trata de las constantes estructuradas, las cuales contienen estructuras de datos, como arrays y records. Una constante literal es aquella que se representa como un valor en el código fuente. Ejemplo:

$$\text{VolumenEsfera} := (4/3)*PI*(Radio*Radio*Radio)$$

En la línea, 4 y 3 son constantes literales, representando los valores de 4 y 3. Sin embargo, hay otro tipo de constante: el identificador PI, el cual debió haber sido declarado en el programa de la siguiente manera:

CONST

$$PI = 3.14159;$$

Se puede usar la constante literal 3.14159 en lugar del PI, pero PI es más corto y hace la expresión más legible. En cuanto a números se refiere, las constantes hacen al código fuente más legible.

Otro uso de las constantes se da cuando se tiene un parámetro que rara vez cambiará en un programa. Tal vez algún día cambie de valor, entonces resultará ventajoso modificar solamente la declaración en lugar de revisar todas las líneas de código para hacer el cambio.

En Pascal estándar las constantes pueden ser solamente de tipos simples y sets. Eso incluye números reales, enteros, bytes, caracteres, strings, sets y booleanos. Las constantes de tipo enumerado individual pueden ser consideradas como tales, aunque se declaran de manera diferente. Ejemplos:

CONST

PI	= 3.14159;	{Real de punto flotante }
Threshold	= -71.47;	{Real negativo }
Direccion	= \$06;	{Valor hexadecimal }
Usa8087	= true;	{Booleano }
Drive	= 'A';	{Caracter }
Revision	= 'V2.16B';	{String }
Respuesta	= 42;	{Entero }
NoRespuesta	= -42;	{Entero negativo }
Afirmacion	= ['S','s'];	{Set }
Nulo	= '';	{String nulo }

La diferencia más evidente entre una constante y una variable es que el valor de la constante se asigna al tiempo de compilación. No se puede asignar un valor a una constante.

```

program MuestraFecha;
uses
  Crt;
const
  NomMes : array[1..12] of string[3] = ('ENE','FEB','MAR','ABR','MAY','JUN',
    'JUL','AGO','SEP','OCT','NOV','DIC');
var
  Dia, Mes, Anno : Integer;

procedure Error;
begin
  writeln;
  writeln ('** ERROR ** Esta fecha no existe');
  halt
end;

begin
  repeat
    write ('Dame la fecha en formato DD MM AA : ');
    readln (Dia, Mes, Anno);
    if (Dia > 31) or (Mes > 12) or (Anno > 99) or
      (Dia < 1) or (Mes < 1) or (Anno < 1) then Error;
    if (Mes = 2) and (Dia > 28) and (Anno mod 4 = 0) then Error;
    writeln ('La fecha es ',Dia:2,' de ',NomMes[Mes],' de 19',Anno);
  until false;
end.

```

TIPOS

Los datos representan información que los programas manipulan. Los datos pueden ser números, caracteres u otros símbolos, como las condiciones de Boole falso y verdadero, conjuntos de conceptos, los días de la semana, o estructuras elaboradas en función de términos simples.

La riqueza de las expresiones con las que Pascal puede tratar los datos lo colocan muy lejos de lenguajes como BASIC y FORTRAN. Versiones modernas de BASIC pueden manejar strings y diferentes tipos de datos numéricos, pero no permiten manejar estructuras construidas a partir de tipos de datos simples.

El "tipo" de un dato es una serie de reglas que gobiernan el almacenamiento y el manejo de cada dato. Los datos ocupan un lugar en el RAM. El tipo de dato determina cuánto espacio es necesario y cómo el dato se representa en ese espacio. Un entero, por ejemplo, siempre ocupa dos bytes de memoria. El bit más significativo representa el signo del entero.

El tipo también determina cómo puede ser usado el dato. Los tipos CHAR y BYTE equivalen a un byte en RAM. Se pueden sumar o multiplicar dos variables de tipo BYTE, pero intentar usar dos variables de tipo CHAR en operaciones aritméticas, genera un error.

La versión 5.0 de Turbo Pascal introduce tipos de datos nuevos, tanto para escalares como para reales. Los tipos de datos escalares nuevos son: word, shortint y longint.

CHAR

El tipo CHAR (caracter) está incluido en el ISO Standar Pascal, por lo tanto, en todas las implementaciones de Pascal.

El tipo CHAR incluye el juego de 128 caracteres ASCII: letras, números, símbolos comunes y los códigos no imprimibles de control, como el retorno de carro, el backspace, etc. Pero actualmente el tipo CHAR incluye 256 caracteres de diferentes valores, ya que ocupan un byte. Los 128 caracteres restantes no tienen nombres ni significados estandarizados en el código ASCII. Cuando se imprimen en la pantalla de la IBM PC, se muestran como letras de idiomas diferentes al Inglés, marcos o símbolos matemáticos. En otras computadoras, estos caracteres se imprimen centelleando o subrayados en variantes del código ASCII estándar.

BYTE

El tipo BYTE no está presente en el ISO Standar Pascal, aunque muchas implementaciones de Pascal lo incluyen. Como CHAR, BYTE ocupa un byte en memoria. A nivel máquina, BYTE y CHAR son iguales, solamente difieren en la forma como son tratados por el compilador. El tipo BYTE puede ser concebido como un entero de "media precisión". Las variables de tipo BYTE solamente pueden compartir asignaciones con el tipo INTEGER.

SHORTINT

El tipo shortint (short Integer) puede ser tomado como un byte con signo. Utiliza un byte para almacenamiento, pero el bit más significativo es el bit de signo. Esto le da a shortint un rango de -128 a 127.

INTEGER

El tipo INTEGER es parte del ISO Standar Pascal. Es la manera más eficiente de manejar números pequeños. Los enteros están en el rango de -32768 a 32767.

El identificador predefinido MAXINT es una constante conteniendo el máximo valor que puede ser expresado: 32767. MAXINT es una característica del ISO Standar Pascal, y existe en el TURBO Pascal para ayudar a los programadores a hacer los programas más portables, ya que otras computadoras manejan palabras de datos de tamaño diferente (computadoras de 32 bits definen a MAXINT como 2,147,483,647).

WORD

El tipo word es llamado algunas veces como un entero sin signo. Como el tipo integer, un word toma 2 bytes. En los enteros se usa el bit más significativo para indicar el signo, mientras que en el tipo word no se utiliza un bit para el signo. Esto le da a word un rango de 0 a 65535.

LONGINT

Es probablemente la adición más significativa de tipos de datos en Turbo Pascal. Longint (long integer) utiliza 4 bytes de almacenamiento y tiene un rango de -2,147,483,648 a 2,147,483,647. En consecuencia, se pueden usar enteros largos en muchos lugares donde antes solo se podía trabajar con reales. Esto es especialmente importante, pues los cálculos que usan enteros toman menos tiempo que los que usan reales.

BOOLEAN

El tipo BOOLEAN es parte del ISO Standar Pascal. Una variable booleana puede tener solo dos posibles valores: True y False (verdadero y falso). Las palabras TRUE y FALSE son identificadores predefinidos en Pascal. El compilador las define como constantes del tipo boolean.

Las variables booleanas se usan para manipular los resultados de expresiones, usando los operadores relacionales =, >, <, <>, >= y <=, así como los operadores de conjuntos +, * y -. Una expresión como "2 < 3" es fácil de evaluar; obviamente 2 es menor que 3 y el resultado es verdadero. Si esto fuese un estatuto en Pascal, la expresión regresaría un valor booleano True, el cual puede ser procesado como sigue:

```
OK := 2 < 3;
```

Esta asignación guarda el valor booleano True dentro de la variable booleana OK. El valor de OK puede ser usado en estatutos IF..THEN..ELSE.

NUMEROS REALES DE PUNTO FLOTANTE

Todos los tipos de datos descritos anteriormente han sido tipos ordinales. Los tipos ordinales tienen un número limitado de valores posibles, existiendo en un orden definido. Los enteros son de tipo ordinal, ya que son exactamente 65535; y tienen un orden definido, ya que después del 6 está el 7, luego el 8, sin valores posibles entre ellos. Los enteros son de precisión absoluta; esto es, el valor del entero 6 es exactamente seis.

REAL

El mundo real demanda fracciones. ISO Standar Pascal soporta el tipo REAL, el cual expresa números con fracciones y exponentes. Los números reales, especialmente los muy pequeños o muy grandes, no tienen precisión absoluta.

Los números reales pueden ser expresados de dos maneras en Turbo Pascal: Una es la mantisa (1.6125), que da el valor, y el exponente (E10), que da el orden de la magnitud. La segunda manera es expresar el número con un punto decimal: 3.14159, 121.402, -16.66, etc.

En adición, Turbo Pascal 4.0 ofrece 5 tipos de reales: real (igual a versiones anteriores), single, double, extended y comp. Desafortunadamente, los tipos single, double, extended y comp solo pueden usarse si la computadora tiene el chip coprocesador matemático 8087.

SINGLE

El tipo single es el de menor precisión de los tipos reales. Utiliza 4 bytes y tiene un rango de 1.5E-45 a 3.4E-38

DOUBLE

El tipo double ocupa 8 bytes y tiene un rango de 5.0E-324 a 1.7E-308

EXTENDED

El tipo extended tiene 10 bytes de longitud y un rango de -2E+63+1 a 2E+63-1

COMP

A pesar de que el tipo de dato COMP es, técnicamente, una variable real, éste actúa como un INTEGER y contiene solamente números sin decimales. Utiliza 8 bytes y tiene un rango de -2E+63+1 a 2E+63-1

```
program DemoNumerico;
```

```
uses
```

```
  Crt;
```

```
var
```

```
  Fin : boolean;
```

```
  arg : real;
```

```
procedure ExpLnDemo; { El procedimiento ExpLnDemo muestra  
                      las funciones Exp y Ln. }
```

```
var
```

```
  i : shortint;
```

```
begin
```

```
  writeln ('** LOGARITMOS **':50);
```

```
  write (' ':6, 'n');
```

```
  write (' ':6, 'Exp (n)');
```

```
  writeln (' ':4, 'Ln (n)');
```

```
  for i := -10 to 10 DO
```

```
  begin
```

```
    arg := 1 / 2;
```

```
    write (arg: 10: 4, Exp (arg): 10: 4);
```

```
    if i <= 0 then writeln ('- ':8) else writeln (Ln(arg):10:4);
```

```
  end
```

```
end; {-----}
```

```
procedure TrigDemo; { El procedimiento TrigDemo muestra el uso  
                    de las funciones trigonométricas: sin,
```