

CAPITULO 1.- CONCEPTOS BASICOS

INTRODUCCION INFORMAL A PASCAL

Una estructura está hecha de algo. La estructura de un cristal consta de átomos arreglados en un orden particular. Un programa en Pascal está hecho de "átomos", los cuales son simples palabras en inglés. Esos átomos del programa caen dentro de tres categorías:

- Un pequeño núcleo de palabras (43 en TURBO Pascal) llamadas "palabras reservadas".
- Un número grande de palabras llamadas "identificadores estándar".
- Un número ilimitado de identificadores ordinarios, creados por el programador.

Las palabras reservadas tienen un significado especial dentro de Pascal. No pueden ser usadas por el programador, a no ser dentro de los límites especificados para su uso. El compilador encontrará error en el uso de una palabra reservada que no se ajuste al uso para el que fue destinada. Ejemplos: BEGIN, END, PROCEDURE, ARRAY, y ese viejo diablo: GOTO.

Todo lo que no sea una palabra reservada, es un "identificador". Algunos identificadores tienen significados predeterminados para el compilador. Esos "identificadores estándar", como las palabras reservadas, tienen un significado particular para el compilador. Sin embargo, bajo ciertas circunstancias, se pueden redefinir los significados para propósitos bastante particulares.

Cualquier nombre aplicado a una entidad en un programa, es un identificador ordinario. Los nombres de las variables, los procedimientos, las funciones y el programa en sí mismo, son identificadores. Un identificador creado por el programador puede servir (dentro de los límites y las reglas de Pascal) para un propósito único. Esto es, si se tiene una variable llamada CONTADOR, no se puede tener un procedimiento o función llamado CONTADOR, ni tampoco otra variable llamada CONTADOR. La siguiente es una revisión a los elementos de un programa de Pascal. El programa en sí mismo debe ser construido con esos elementos, de una muy particular manera. La razón de ello es proporcionar al programador una visión sencilla de la tarea del programa, en forma clara y estructurada.

ESTRUCTURA DEL PROGRAMA

EL ESTATUTO PROGRAM.

El estatuto PROGRAM le da al programa un nombre y le dice al compilador en dónde empieza el programa. El nombre del programa no es necesariamente igual al nombre del archivo que lo contiene, ni tampoco igual al nombre del archivo que contiene el código ejecutable que el compilador hace como producto final. Este es un estatuto program:

```
PROGRAM FIME;
```

EL ESTATUTO USES

Una vez que se desarrolló y compiló una unidad, es necesario incorporar las rutinas que contienen la unidad dentro del programa. El estatuto USES es el medio para hacerlo. Debe declararse al inicio del programa, después de la declaración del identificador del programa. La sintaxis es:

```
USES Unidad1, Unidad2, Unidad3...;
```

Los nombres de la lista de unidades deben ser los de unidades ya creadas y que se encuentran en disco, por ejemplo UNIDAD1.TPU, UNIDAD2.TPU, etc. Cuando el compilador encuentra el estatuto USES, busca en el disco la unidad e incorpora las rutinas públicas en el programa. encadenar las rutinas con el programa a compilar: Únicamente las rutinas que usa el programa son las que se convierten en código en el producto final. Por ejemplo, si una unidad tiene diez rutinas públicas, pero el programa que la utiliza sólo manda llamar a tres, Turbo Pascal encadena estas tres solamente.

Si se compila un programa que involucra varios archivos de código, Turbo Pascal es muy eficiente al

El siguiente programa es un ejemplo de la declaración USES. Se asume la unidad "PRUEBA", compilada y lista para usarse en disco, y que incluye la rutina "IMPRIME":

```
program PruebaUnidad;
uses
  Crt, Prueba;           { incluye las unidades CRT y PRUEBA }
begin
  clrscr;                { incluye la función de CRT }
  writeln;
  Imprime ('Hola !');    { utiliza la función de PRUEBA }
end.
```

LA SECCION DE DECLARACION DE ETIQUETAS (LABEL)

La parte siguiente es la sección de declaración de etiquetas. Consiste en la palabra reservada LABEL, seguida de una lista de etiquetas. Las etiquetas son marcas en un programa, al cual debe ir el estatuto GOTO. Deben contener números no- negativos, no importando su orden ni su rango, pero cada una debe ser única. Este es un ejemplo de una declaración de etiquetas:

```

LABEL
    50,100,150,200;
    
```

Si no se tienen GOTO's en el programa (y ésta es una muy buena idea), no necesitará etiquetas. No tener etiquetas significa que la parte de la declaración de etiquetas es innecesaria y puede omitirse. Pero si se tiene por lo menos una, la declaración debe ser el primer estatuto del programa.

LA SECCION DE DECLARACION DE CONSTANTES (CONST)

Si se intenta usar constantes en un programa, el compilador las localiza en la sección de la declaración de constantes. Los valores de las constantes son definidos al tiempo de compilación y nunca son cambiados en el proceso de ejecución del programa. La declaración de las constantes consiste en la palabra reservada CONST, seguida de la lista de constantes. La declaración de una constante consiste en un identificador y su valor, separados por un signo de igual (=).

```

CONST
    Limite      = 255;
    GNombre     = 'Gráfica 3-D';
    Plotter     = true;
    
```

En el ejemplo anterior, LIMITE es una constante entera; GNombre es una constante string y Plotter una constante booleana. El Pascal estándar y el TURBO Pascal difieren significativamente por la forma en que puede ser una constante. Esto se verá posteriormente.

LA SECCION DE DECLARACION DE TIPOS (TYPE)

La declaración de tipos consiste en la palabra reservada TYPE, seguida por la lista de tipos. Una declaración de tipos es un tipo de identificador y su declaración, separado por un signo igual (=). Pascal predeclara los tipos fundamentales: entero (integer), booleano (boolean), caracter (char), byte (byte) y real (real). Estos no necesitan ser declarados. Solo se declaran aquellos que son creados por el usuario, o los que son subrangos de los tipos fundamentales ya definidos.

Se pueden definir nuevos tipos a partir de tipos ya definidos por el usuario; por lo tanto, no se puede usar un tipo como ingrediente de un nuevo tipo, a menos que el compilador conozca de qué tipo se trata. El compilador conoce un tipo cuando éste ya ha sido predeclarado (integer, byte, etc), o porque ya ha sido compilado el estatuto definiendo el tipo en términos de otros tipos ya definidos con anterioridad.

```

TYPE
    Grupo      = 'A'..'G';           {subrango}
    Llamada     = string[8];
    USHam      = record
        LlamadaOp :Llamada;
        Clase     :char;
    end;
    Conta      = record
        SuLlamada :Llamada;
        RST       :integer;
        Banda     :integer;
        Frecuencia :real;
        Emision   :string[2];
        Tiempo    :integer;
    end;
    Inicio      = record
        NuestraOp :USHam;
        Equipo    :Grupo;
        QSO       :Conta;
    end;
    
```

La anterior es una serie de declaraciones que podrían aparecer en un programa que manejase un radio de CB. Si los detalles de la declaración son extraños para ud., no se preocupe, serán cubiertos en forma minuciosa posteriormente. Lo importante es notar que todos los tipos están definidos en función de los tipos fundamentales o de subrangos (a través del signo =), antes de que sean usados. Por ejemplo, todos los tipos que forman Inicio son definidos por el usuario. Cada uno debe tener una definición anterior a Inicio, y de hecho cada uno la tiene. Si Grupo fuese definido después de Inicio, el compilador marcaría error durante la compilación del programa. No sabría qué almacenar al encontrar el primer Grupo. Grupo es un subrango de tipo char (que incluye las letras de la A a la G), y una vez definido en esos términos, el compilador ya sabe a qué se refiere. Así como la parte de etiquetas, la declaración es opcional; si no hay tipos que definir, puede ser omitida.

Ordinariamente, la declaración de constantes es anterior a la de tipos, pero TURBO Pascal permite declarar tipos antes que constantes, si así se desea. Esto se hace en conjunto con las constantes de arreglos (arrays) y las de registro (record).

LA SECCION DE DECLARACION DE VARIABLES (VAR)

Estrictamente hablando, un programa no necesita una sección de declaración de variables para ser compilado o ejecutado. Sin embargo, a pesar de que un programa puede carecer de las secciones de declaración de tipos y de constantes, no se podría hacer gran cosa sin algunas variables para trabajar con ellas.

La declaración de variables consiste en la palabra reservada VAR, seguida por la lista de variables. Cada variable usada debe tener un tipo válido, ya sea predefinido (integer, real, boolean, etc) o definido por el usuario en la parte de declaración de tipos del programa. A diferencia de la declaración de tipos, la declaración de variables utiliza el símbolo de dos puntos (:) para separar la variable de su tipo:

```
VAR
  Contador :integer;
  Destino  :char;
  Zona    :puntoXY;
```

Si algunas variables tienen el mismo tipo, pueden ser agrupadas en una sola línea, separándolas por comas:

```
VAR
  I,J,K : integer;
```

PROCEDIMIENTOS (PROCEDURE) Y FUNCIONES (FUNCTION)

Después de que todas las variables han sido declaradas, se deben definir todos los procedimientos y funciones que el programa va a necesitar. Los procedimientos y las funciones son las subrutinas en Pascal. Tienen la forma de un programa en miniatura e idéntica estructura que el programa principal, excepto que principian con la palabra reservada PROCEDURE (o FUNCTION, según sea el caso) en lugar de PROGRAM. No se necesita un punto, sino punto y coma para terminarlos. Las funciones, adicionalmente, regresan un valor.

El resto de la estructura es la misma. Procedimientos y funciones pueden tener sus declaraciones "privadas" de etiquetas, constantes, tipos definidos por el usuario, variables; e inclusive pueden tener sus propios procedimientos y funciones. Este anidamiento podría seguir hasta el infinito, pero aquí es donde están las restricciones de la implementación sobre la que se esté trabajando en particular.

El compilador de Pascal fuerza las llamadas a los procedimientos y las funciones en forma jerárquica. Un procedimiento o función puede llamar a cualquier procedimiento o función que se haya declarado anteriormente.

EL PROGRAMA PRINCIPAL

El programa principal, que se encuentra al final del archivo fuente, es el "jefe" que llama a cualquier procedimiento. Como se dijo anteriormente, el cuerpo principal del programa es tratado por el compilador en forma similar a los procedimientos y las funciones, excepto que siempre finaliza con un punto.

Cuando el programa es ejecutado, "entra" al programa principal en el BEGIN y no desde el principio del archivo, como podría pensarse. Cuando encuentra el END del programa principal, abandona éste para regresar al sistema operativo. Mientras lo hace, sigue el flujo de control del programa, llamando a todos los procedimientos y funciones que se indiquen en él.

IDENTIFICADORES

La computadora crea programas para su uso. Los programas son un conjunto de datos y pasos para almacenar, modificar o desplegar datos. La computadora conoce cada parte de los datos, porque ellos tienen una dirección en la memoria. Los nombres propios que se le asignan a los datos, procedimientos y programas, son para beneficio del programador. A estos nombres se les denomina "identificadores", y existen solamente en el código fuente. Los identificadores no existen en el código final.

Los identificadores son secuencias de caracteres de cualquier longitud, hasta 127, y deben seguir estas reglas:

1. Los caracteres legales son las letras, dígitos y el símbolo de subrayado (_). Espacios y símbolos como &, !, *, % no se permiten.
2. Símbolos o dígitos (0 al 9) no pueden ser usados como primer carácter. Todos los identificadores deben empezar con una letra (mayúscula o minúscula) o con el subrayado.
3. Los identificadores no deben ser iguales a ninguna palabra reservada.
4. Las diferencias entre mayúsculas y minúsculas son ignoradas. Para el compilador, "a" es igual a "A".
5. Los subrayados son legales y significantes. Nótese que difieren de algunos otros compiladores de Pascal, en donde los subrayados son legales pero se ignoran. Los subrayados se utilizan para hacer más legible el identificador. Por ejemplo, CODIGOPOSTAL y CODIGO_POSTAL. Otro método es: CódigoPostal.
6. Todos los caracteres son significantes, no importa qué tan grande sea el identificador, siempre y cuando no exceda los 127 caracteres. Muchos otros compiladores de Pascal ignoran todos los caracteres después del octavo.