Sobrecarga

Una posibilidad muy interesante que tenemos en Delphi es la de sobrecarga de procedimientos o funciones.

La sobre carga se produce cuando definimos una función o un procedimiento de distintas formas (mismo nombre de función distintos parámetros y código).

En muchas ocasiones necesitamos una función de devuelva un valor de un tipo u otro dependiendo del tipo de parámetros que la pasemos, pero conservando la filosofía de la función, un caso significativo es la función 'Iif' (if condición then result:=valor1 else result:=valor2) que aparecen en otros lenguajes.

¿Qué ocurre si intentamos definir dos funciones (y/o procedimientos) con el mismo nombre? El compilador se queja. Para que esto no ocurra existe en delphi la directiva '***overload***'.

Pero pasemos a un ejemplo, vamos a definir una función Iif que nos devuelve un valor u otro dependiendo de que se cumpla una condición o no.

```
Unit NewFunc;

interface
uses
  ......
type
  ...........
function Iif(cond:boolean;ctrue:string;cfalse:string):string; overload;
function Iif(cond:boolean;ctrue:integer;cfalse:integer):integer; overload;
function Iif(cond:boolean;ctrue:double;cfalse:double):double; overload;
function Iif(cond:boolean;ctrue:boolean;cfalse:boolean):boolean; overload;
...............
implementation
...............
// La siguiente función devuelve una cadena dependiendo de la condición
function Iif(cond:boolean;ctrue:string;cfalse:string):string;
begin
   if cond then result:=ctrue else result:=cfalse;
end;
// La siguiente función devuelve un entero dependiendo de la condición
function Iif(cond:boolean;ctrue:integer;cfalse:integer):integer;
begin
   if cond then result:=ctrue else result:=cfalse;
end;
// La siguiente función devuelve un  double dependiendo de la condición
function Iif(cond:boolean;ctrue:double;cfalse:double):double;
begin
   if cond then result:=ctrue else result:=cfalse;
end;
// La siguiente función devuelve un boolean (true/false) dependiendo de la condición
function Iif(cond:boolean;ctrue:boolean;cfalse:boolean):boolean;
begin
   if cond then result:=ctrue else result:=cfalse;
end;
.................
end.
```

Ejemplos :

```
Var
    A: string;
    B: Double;
    C: Integer;
Begin
```

```
   A:=iif(vari='P','Pepe','Juan');
   B:=iif(vari='P',2.3,1,23);
   C:=iif(vari='P',1,2);
End;
```

Si vari vale P entonces :
A vale 'Pepe'
B vale 2.3
C vale 1

Pero las funciones y/o procedimientos sobrecargados no tienen por qué tener el mismo número de parámetros como vamos a ver en el siguiente ejemplo.

Vamos a definir una función del tipo Decode de Oracle, esta función es una especie de 'Case' en el cual dependiendo de el valor de un parámetro devolverá un valor. Ejemplo en oracle :

Decode(valor, 'pepe', 'lopez','juan','sanz','lola','diaz','apellido desconocido')

Esta funcion devuelve :

Valor	valor devuelto
pepe	lopez
juan	sanz
lola	diaz
<resto>	apellido desconocido

En nuestro caso vamos a definir esta función para cadenas (string) de dos formas distintas, sobrecargando la función :

```
Unit NewFunc1;

interface
uses
  ......
type
  ..........
// Con cuatro parámetros
function Decode(cond:string;valores:array of string;results:array of string;
ifnoexist:string):string;overload;
// Con tres parámetros
function Decode(cond:string;valores:array of string;
ifnoexist:string):string;overload;

.................
implementation
.................
function Decode(cond:string;valores:array of string;ifnoexist:string):string;
var i:integer;
begin
   result:=ifnoexist;
   i:=low(valores);
   while i<high(valores) do
   begin
      if valores[i]=cond then
      begin
         result:=valores[i+1];
         break;
```

```
      end;
      i:=i+2;
   end;
end;
function Decode(cond:string;valores:array of string;results:array of string;
ifnoexist:string):string;
var i:integer;
begin
   result:=ifnoexist;
   for I:=low(valores) to high(valores) do
      if valores[i]=cond then
      begin
         result:=results[i];
         break;
      end;
end;
```

Ejemplos :

```
Var
    A: string;
    B: String;
Begin
   A:=decode(vari,['PP','QST','WRD'],['Pepe','Quest','Word'],'Nose');
   B:=decode(vari,['PP', 'Pepe', 'QST', 'Quest', 'WRD', 'Word'], 'Nose');
End;
```

Esta funcion devuelve en los dos casos:

Valor	valor devuelto
PP	Pepe
QST	Quest
WRD	Word
<resto>	Nose

Hay que tener cuidado cuando definimos una función sobrecargada en la cual uno de los parámetros tiene un valor por defecto poeque la definición podría ser ambigua :

```
Procedure Pp(a:integer;b: char = 'B');overload;
Procedure Pp(a:integer);overload;

Begin
   Pp(2);
End;
```

Esto daría un error, ya que al tener el parámetro b por defecto el valor 'B', el compilador no sabría cual es el procedimiento al que tiene que invocar, para solucionarlo basta con quitar el valor por defecto.