

Tema 9

Tipos Definidos: Enumeración Y Conjuntos

Definición de tipos

Mediante la definición de tipos de datos por el programador se consigue que cada información que maneja el computador tenga su sentido específico. El tipo establece los posibles valores que puede tomar ese dato. Además, al igual que sucedía con los tipos predefinidos, a cada nuevo tipo que se define se asocian un conjunto de operaciones que se pueden realizar con él. Por tanto, la definición de tipos supone crear un nuevo nivel de abstracción dentro del programa.

Tipos enumerados

- **Definición de tipos enumerados:** Para crear un nuevo tipo de dato se enumeran todos los posibles valores que puede tomar separados por comas (,) y encerrados entre paréntesis. Cada posible valor se describe mediante un identificador (que al mismo tiempo quedan declarados como valores constantes).

```
TpoEnum = (id1, id2, ..., idN);
```

La enumeración implica un orden que se establece entre los valores enumerados. Este orden se define de forma implícita e impone que el primer elemento de la lista ocupa la posición 0, el siguiente la 1, y así sucesivamente hasta el último, que ocupa la posición N-1, siendo N el número de elementos enumerados.

Forman parte de la clase de tipos ordinales: INTEGER, CARDINAL, CHAR pero **no el REAL**.

- **Uso de tipos enumerados:** Los tipos enumerados se emplean de manera similar a los tipos predefinidos. El identificador de tipo se puede emplear para definir variables de ese tipo, y los identificadores de los valores enumerados se emplean como las constantes con nombre.

Función	Uso
ORD(X)	Posición que ocupa X en la lista de su enumeración
VAL(T, X)	X convertido al tipo T
INC(X [, N])	Incrementa en 1/N el valor de la variable X
DEC(X [, N])	Decrementa en 1/N el valor de la variable X

Un dato de tipo enumerado se puede pasar como argumento de procedimientos o funciones y puede ser el resultado de una función.

El tipo predefinido BOOLEAN

El tipo predefinido BOOLEAN es el identificador del tipo, y las constantes simbólicas TRUE y FALSE corresponden a los valores de verdad *cierto* y *falso*, respectivamente.

```
TYPE BOOLEAN = (FALSE, TRUE);
```

Operador	Operación lógica
AND &	Conjunción
OR	Disyunción
NOT ~	Negación

Tipos subrango

Otra forma de especificar nuevos tipos de datos es estableciendo rangos parciales de valores de otro tipo ordinal (**no el REAL**) ya existente. Con esto no se definen nuevos valores, pero sí un nuevo tipo con una colección limitada de valores válidos.

- **Definición de tipos subrango:** Un tipo subrango se define a partir de otro tipo ordinal ya definido, que se toma como *tipo base*. Para ello se declara un identificador diferente para el nuevo tipo y se establecen los límites mínimo (*primero*) y máximo (*último*) del subrango de variación separados por dos puntos seguidos (..) y se encierran entre corchetes ([]).

```
TpoSub = [primero..último];
```

O con el *tipo base*:

```
TpoSub = TpoBase[primero..último];
```

- **Uso de tipos subrango:** Las variables de tipo subrango tienen la misma consideración que las variables de su *tipo base*. Por tanto, se pueden utilizar en expresiones o sentencias de asignación como si fueran variables del *tipo base*.

Tipos estructurados

Los tipos de datos vistos hasta el momento se denominan *escalares*, y son datos simples, en el sentido de que no se pueden descomponer.

Un tipo estructurado de datos, o estructura de datos, es un tipo cuyos valores se construyen agrupando datos de otros tipos más sencillos. A sus elementos se les denominan *componentes*. Todos los tipos estructurados se definen a partir de tipos simples combinados.

Conjuntos

El concepto matemático asociado a elementos de información de esta clase es la noción de **CONJUNTO**. Los valores posibles conjunto son los *subconjuntos* de un *conjunto referencial* que contiene todos los valores posibles.

- **Definición de tipos conjunto:** Se realiza tomando como base o conjunto referencial el conjunto de todos los valores posibles de un tipo ordinal (**no el REAL**) definido previamente. Por analogía con el vocabulario de conjuntos, llamaremos *tipo referencial* a este tipo base.

```
TpoConj = SET OF TpoReferencial;
```

- **Construcción de conjuntos:** Para asignar valores a las variables se utiliza una expresión de construcción de conjunto, en la que se enumeran, entre llaves ({}), los elementos del conjunto a incluir, separados por comas (,) y precedidos por el identificador del tipo de conjunto al que pertenecen los elementos.

```
variable1 = TpoConj{ val1, val3 };
variable2 = TpoConj{};
variable3 = TpoConj{ val1..val5 };
```

Si no se indica ningún elemento dentro de los corchetes, el conjunto asignado es el conjunto vacío. La lista de elementos puede ser tan larga como sea necesario y el orden en que se relacionan es irrelevante. Para abreviar, si hay varios elementos consecutivos basta con indicar el primero y el último separados por dos puntos seguidos (..).

- **Operaciones entre conjuntos:**

Operación	Uso
$X + Y$	Unión: elementos de X , o de Y , o de ambos
$X * Y$	Intersección: elementos comunes a X e Y
$X - Y$	Diferencia: elementos de X que no son elementos de Y
X / Y	Diferencia simétrica: elementos de X o de Y pero no de ambos

Esta es la lista de funciones que se pueden utilizar con los conjuntos:

Función	Uso
EXCL(S, X)	Excluye el elemento X del conjunto S
INCL(S, X)	Incluye el elemento X en el conjunto S

Estos operadores devuelven un valor lógico:

Operadores	Uso
$X = Y$	Equivalencia: X e Y tiene los mismos elementos
$X < > Y$	Desigualdad: X e Y tienen algún elemento distinto
$X \leq Y$	Inclusión: todos los elementos de X pertenecen a Y
$X \geq Y$	Inclusión: todos los elementos de Y pertenecen a X
elem IN conj	Permite comprobar si un elemento elem pertenece al conjunto conj

Nota: el conjunto vacío siempre esta incluido en cualquier otro.