

TIPOS DE DATOS

Explicación Práctica

Programación I - 2024

Facultad de Informática y Facultad de Ingeniería - UNLP

Tipos de datos

Un **tipo de dato** define un conjunto de valores permitidos y una serie de operaciones sobre estos valores.

En Pascal tenemos dos grupos de tipos:

Predefinidos

- Integer
- Char
- Boolean
- Real

Definidos por el usuario

- Subrango
- String
- Conjuntos

Tipos definidos por usuario

Declaración

La declaración se hace a través de la palabra clave
TYPE

```
program ejemplo;  
type  
    nombreTipo = declaracionTipo;  
  
var  
    {Declaración de variables del programa  
    principal}  
  
begin  
    {Sentencias}  
  
end.
```

Subrango

Subrango: Sucesión de valores de un tipo **ordinal** definido previamente (*char*, *integer*) con valores consecutivos en orden **creciente**.

Las operaciones permitidas se heredan del tipo base.

Ejemplos de declaración

TYPE

```
meses = 1..12;  
letras = 'a'..'z';
```

VAR

```
m1,m2,m3: meses;  
l: letras;
```

Ejemplos de operaciones

BEGIN

```
readln(m1); //Posible error  
m1:= 4;  
m2:= 9;  
m3:= m1+m2; //Error:fuera de  
          rango  
writeln(m2):
```

EN

Ventaja: validación
de datos

String

String: N caracteres consecutivos tratados como una única variable.

Ejemplos de declaración

TYPE

```
cadena = string[10];  
generico = string;
```

VAR

```
s: cadena;
```

Máxima longitud permitida para un string: **255 caracteres**

Ejemplos de operaciones

- Asignación
- Comparación
(=, <>, >=, <=, <, >)
- Read()
- Write()

Conjuntos

Conjuntos: Colección homogénea de elementos, sin repetición, sin relación de orden, e “ilimitada”.

En Pascal:

Se pueden tener conjuntos de **tipos ordinales** (*enteros, caracteres, subrango*).

El número máximo de elementos, por cuestiones de implementación, está limitado a **255**.

Conjuntos - Operaciones

NO SE PUEDE HACER
READ/WRITE

Operación	Operador	Tipo de resultado
Unión	$+$	Conjunto Ej: $C := A + B$
Intersección	$*$	Conjunto Ej: $C := A * B$
Diferencia	$-$	Conjunto Ej: $C := A - B$
Pertenencia	IN	Boolean Ej: if(ELEM in A)then
Comparación	$A \neq B$	Boolean (<i>True si los conjuntos A y B son distintos</i>)
	$A \leq B$	Boolean (<i>True si A está incluido en B</i>)
	$A \geq B$	Boolean (<i>True si B está incluido en A</i>)
	$A = B$	Boolean (<i>True si A y B son iguales</i>)

Conjuntos - Declaración

TYPE

```
meses = 1..12;  
conj_meses = set of meses;  
conj_letras = set of char;  
conj_nros = set of integer;
```

VAR

```
c1,c2 : conj_meses;  
conjMin, conjMay : conj_letras;
```

BEGIN

```
c1 := [5];  
c2 := [1..3, 9];  
c1:= c1 + c2;  
c1:= c1 + [10];  
conjMin := ['a'..'z'];  
conjMay := ['A' .. 'Z'];  
if (10 in c1) then  
  writeln('10 esta en c1');
```

1)
Declaración
del tipo

2)
Declaración
de la
variable

3)
Inicialización
del conjunto

**Illegal en
Pascal**

No puede tener
mas de 255
elementos. Lo
permitiremos en
la práctica en
papel.

Ejercicio

Se leen 5 números desde teclado. Informar cuáles son los dígitos que **no** aparecieron.

Ejemplo, si se lee:

212 34 418 909 9224

Debería informar que no aparecieron los dígitos:

5, 6, 7

Solución

```
Program ej;
Type
    Tdigitos = 0..9;
    Tconj = set of Tdigitos;
Var
    i, num, resto : integer;
    conj: Tconj;
Begin
    conj:=[];

    for i:= 1 to 5 do begin
        readln(num);
        {Descomponemos num, poniendo cada dígito en el conjunto}
        while (num <> 0) do begin
            resto := num mod 10;
            conj:= conj + [ resto ];
            num := num div 10;
        end;
    end;

    writeln('Los dígitos que NO aparecieron son:');
    for i:= 0 to 9 do
        if not (i in conj) then {No apareció}
            writeln(i);
    end.
end.
```

Ejercicio

Se leen 5 números desde teclado. Informar cuáles son los dígitos que no aparecieron y el dígito mayor de cada número leído.

¿Qué modificaciones deberíamos hacer en el ejercicio anterior?

Program ej;

Type

Tdigitos = 0..9;

Tconj = set of Tdigitos;

Var

i, num, resto: integer;

max, num_copia : integer;

conj: Tconj;

Begin

conj:=[];

for i:= 1 to 5 do begin

 readln(num);

 max:= -1;

 num_copia:= num;

 while (num <> 0) do begin *{Descomponemos el nro}*

 resto := num mod 10;

 conj:= conj + [resto];

 if (resto > max) then

 max:= resto;

 num := num div 10;

 end;

 writeln('El digito mayor del numero', num_copia, 'es', max);

end;

writeln('Los digitos que NO aparecieron son:');

for i:= 0 to 9 do

 if not (i in conj) then *{No apareció}*

 writeln(i);

end.

Solución