

Slide 1

TEMA 7: OTROS TIPOS Y ESTRUCTURAS DE CONTROL (PASCAL)

Índice

7.1. Tipos de datos simples definidos por el programador .	1
7.1.1. Tipo enumerado	1
7.1.2. Tipo subrango	3
7.2. Tipos con nombre	4
7.3. Estructuras de control adicionales	5
7.3.1. Condicional múltiple sobre ordinal	5
7.3.2. Estructura iterativa con índice	7

Slide 2

TIPO ENUMERADO

Tipo simple, ordinal, definido por el programador.

valores: los que el programador defina:

(<identificador> { , <identificador>})

operaciones: operadores relacionales (resultado **boolean**):

menor que: < menor o igual: <= igual: =

mayor que: > mayor o igual: >= distinto: <>

funciones: predecesor: **pred** sucesor: **succ**
número de orden : **ord**

No están contempladas lectura de teclado ni escritura en pantalla.

Los identificadores deben ser únicos.

Slide 3

```

var mes1, mes2 : (ene, feb, mar, abr, may, jun,
                 jul, ago, sep, oct, nov, dic);
    estado      : (soltero, casado, viudo, separado,
                 emparejadodehecho, clerigo);
    color       : (rojo, amarillo, azul, naranja,
                 verde, violeta);
    diasem      : (lunes, martes, miercoles, jueves,
                 viernes, sabado, domingo);
    nota        : (ut, re, mi, fa, sol, la, si);
Begin
  if (dia=13) and (diasem = martes) then estado := casado;
  if mes1 = dic then mes2 := ene
  else mes2 := succ(mes1);
  if diasem > viernes then (** fin de semana **)
  ...

```

Slide 4

TIPO SUBRANGO

Tipo simple, ordinal, definido por el programador.

valores: subconjunto elementos consecutivos de un tipo ordinal:

<primer valor> .. <último valor>

siendo *<primer valor>* ≤ *<último valor>*

operaciones y funciones: las que el tipo del que se extrae permita.

Se detectará la permanencia en el rango (quizá mediante una opción de compilación). (Opción -Cr ó {\$R+})

Slide 5

```

                                TIPOS CON NOMBRE

type <identificador>= <tipo> { ; <identificador>= <tipo>}

type entero = integer;
      tDiasDeUnMes = 1 ..31;
      tDiasDeSemana = (lunes, martes, miercoles, jueves,
                      viernes, sábadó, domingo);
      tDiaLectivo = lunes .. viernes;
      tMinuscula = 'a'..'z';
      tDigito = 0 ..9;
      tCharDigito = '0'..'9';
      tPositivo = 0 .. maxint;
var dia1, dia2 : tDiasDeUnMes; (* entero *)
      dia       : tDiaLectivo; (* enumerado *)

```

Slide 6

```

                                CONDICIONAL MÚLTIPLE

case <expresión ordinal> of
  <elemento lista case>
  {; <elemento de lista case> }
  <parte else optativa (NO ESTÁNDAR)>
end

<elemento de lista case> ::=
  <lista de rótulos de case> : <sentencia> | <vacía>
<lista de rótulos de case> ::= <rótulo case> {,<rótulo case> }
<rótulo de case> ::= <constante> | <constante> .. <constante>
<parte else optativa> ::= ; else <sentencia> |

```

Slide 7

```

Program DiasDeMes (input, output);
type tnatural = 1 .. maxint ;
   tmeses = 1 .. 12 ;
var mes: tmeses; anyo : tnatural ;
BEGIN
  write (output, 'Mes y año:'); readln (input, mes, anyo);
  case mes of
    1,3,5,7..8,10,12 : writeln (output, '31');
    4,6,9,11       : writeln (output, '30');
    2               : if (anyo mod 4 = 0) and (anyo mod 100 <>0)
                      or (anyo mod 400 = 0)
                      then writeln (output, '29')
                      else writeln (output, '28')
  end (* case *)
END.

```

Slide 8

```

                ITERACIÓN CONTROLADA POR ÍNDICE ORDINAL
<identVar>:= <expInicial>;
while <identVar><= <expFinal> do
  begin
    <sentencia>;
    <identVar>:= succ(<identVar>)
  end

```

```

for <identVar>:= <expInicial> to <expFinal> do
  <sentencia>

```

Slide 9

ITERACIÓN CONTROLADA POR ÍNDICE ORDINAL

```
<identVar> := <expInicial> ;  
while <identVar> <= <expFinal> do  
  begin  
    <sentencia> ;  
    <identVar> := pred (<identVar> )  
  end
```

```
for <identVar> := <expInicial> downto <expFinal> do  
  <sentencia>
```