

Capítulo 7

1. Registro. Declaración del tipo.
2. Acceso. Lectura, escritura operaciones sobre Registros.
3. Registros como parte de otra estructura. Definiciones anidadas. Ejemplos.

1. Registro

Tipo estructurado que permite almacenar un conjunto de valores de distinto tipo. Cada una de las componentes del registro se denomina campo, puede ser de tipo simple o estructurado

1.1. Declaración del tipo

```
TipoReg = Record
  idcampo1 : tipo1;
  idcampo2 : tipo2;
  ....
End;
```

Ejemplo de un registro dónde se almacenan los datos de una fecha:

Type

```
TregF = record
  Mes, Dia: byte;
  Anio : word;
end,
```

30	8	2004
Dia	Mes	Anio

Ejemplo de un registro dónde se almacenan los datos de una persona:

Type

```
St20 =string[20];
TregP = Record
  Nombre : St20;
  Edad : byte;
  Peso, Altura : Real
```

'Pía'	20	50	1.6
Nombre	Edad	Peso	Altura

End;

Var

```
Per1, Per2 : TregP;
```

2.1. Acceso

Para acceder a la información almacenada en un registro: *Vble_tipo_registro.Campo*

Como es un tipo estructurado, las operaciones se realizan sobre los campos y no sobre la totalidad de la variable → ~~Readln(Per1);~~

```
Readln(Per1.Nombre);
Readln(Per1.Altura);
-----
```

}

```
Per2.Peso := 50;
Per1.Peso := Per2.Peso;
Per2 := Per1; {se permite asignación entre registros si son del mismo tipo}
Writeln(Per2.Altura)
```

With Per1 Do

Begin

```
Readln(Nombre); Readln(Altura);
```

End;

No se permite la comparación entre registros.

2.2. Campos de tipo estructurado : registros y arreglos

- **Campo registro** Ejemplo: Fecha de nacimiento. *Escribir el año de nacimiento*

Type

TregPer = **Record**

Nombre : St20;

FechNac : TregF; {registros anidados}

End;

Var

Per: TregPer;

.....

Write ('Nació en el año', Per. FechNac.Anio)

- **Campo arreglo** Ejemplo: 5 mediciones de peso (5 meses). *Leer la 3ra medición*

Type

TregPers = **Record**

Nombre : St20;

Pesos : array[1..5] of real; {el tipo arreglo podría definirse previamente}

End;

Var

Pers: TregPers;

.....

readln (Pers.Pesos[3]);

3. Registros como parte de otra estructura

3.1. Arreglo de registros

Las componentes de un arreglo pueden ser de tipo registro. Es posible reemplazar un conjunto de arreglos paralelos por un arreglo de registros (un campo por cada arreglo).

Ejemplo: Se desea almacenar los datos de un conjunto de personas es posible armar un arreglo de registros, donde en cada uno de ellos se puede almacenar los datos de una persona.

.....

'Pía'	20	50	1.6																
	1				2					3						4			

TV = **array**[1 .. MaxElem] **of** TregP;

3.1.1. Lectura de un arreglo, no se conoce la cantidad de elementos:

Procedure LeeVector (Var V:TV; Var N:Byte);

Var

Res:char;

Begin

N:=0; Write('Ingresa datos (S/N)); Readln(Res);

While (Res = 'S') and (N<MaxElem) do

Begin

N:=N+1;

With V[N] **do**

Begin

Readln(Nombre); Readln(Edad);

Readln(Peso); Readln(Altura);

End;

Write('Ingresa datos (S/N)); Readln(Res);

End;

End;

3.1.2.Consulta el Peso a partir del Nombre

Procedure Consulta (V:TV ; N:Byte ; Nom:St20 ; Var Peso:Real);

Var

J : Byte;

Begin

J:=1;

While (J<=N) and (V[J].Nombre<>Nom) **do**

J:=J+1;

If J<=N **then**

Peso:=V[J].Peso

else

Peso:=0;

End;

Actividad

- Hallar el nombre de la persona más joven.
- Hallar el promedio de altura.
- Generar otro arreglo con los nombres y edades de los pacientes con peso normal
 $IMC = \text{Peso} / \text{Altura}^2$; peso normal $18.5 \leq IMC < 25$

3.2.Definiciones anidadas.

Dadas las siguientes declaraciones de tipo y variable:

Type

TRF = Record

dd,mm: byte;

aa: word;

end;

TRExam = Record

Fecha: TRF;

Nota: real;

end;

TRMateria = Record

Codigo: String [4];

Examenes: array [1..10] of TRExam;

ContE: byte;

end;

TRAlumn = Record

Matricula: string [4];

Materias: array [1..20] of TRMateria;

end;

TVALum = array [1..100] of TRAlumn;

Var

V:TVALum

Se pide:

- Escribir del alumno i, la materia j, el mes del examen k.
- asignar nota 8 al último examen de la segunda materia del tercer alumno