



Introducción a la Programación

Grado en Ingeniería Informática

11. Tipos de datos compuestos

Bloque 4. Recorrido y búsqueda
11. Tipos de datos compuestos

Tipos de datos compuestos

Se pueden crear tipos de
datos compuestos a partir de
los primitivos

Los más habituales son:

- Registros (*estructuras*)
- Tablas (*arrays*)
- Secuencias (*listas*)

Registros

Estudiante < dni, nombre, centro, curso >

e1 < 45394545, Pedro, Inf, 1 >

e2 < 34564492, Lola, Mat, 2 >

Tablas

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

Acceso indexado

Secuencias

Sec1 = [28, 33, 56, 1, 87, 44, 67, 12, 98, 09]

Sec2 = [V, F, F, F, V, F, F, V, V,]

Acceso secuencial

Registros / Estructuras

3

Registros / Estructuras

Sus valores son una agrupación de otros tipos ya definidos

Sintaxis

```
nombreTipo = estructura < t1 v1; t2 v2; ... tn vn >  
nombreVariable : nombreTipo
```

Ejemplo

```
T_fecha = estructura < int día, int mes, int año>  
pedro, juan : T_fecha;
```

```
pedro <12, 3, 2020>  
juan.día ← pedro.día;
```

```
“.” : acceso a un campo de un registro
```

4

Estructuras en C/C++

Nombre dado al tipo de la estructura

```
struct nombreTipo  
{
```

Nombres de las variables básicas

```
    tipoBase variable;  
    tipoBase variable;
```

Tipos de datos básicos

```
    ...  
    tipoBase variable;
```

```
} variable, variable;
```

Nombres de las estructuras (opcional)

```
nombreTipo variable, variable;
```

5

Estructuras en C/C++

```
main()  
{  
    struct T_fecha  
    {  
        int dia;  
        int mes;  
        int anyo;  
    };  
  
    T_fecha pedro,juan;  
    int i,j,k;  
  
    pedro.dia=12;  
    pedro.mes=2;  
    pedro.anyo=2020;  
  
    juan.dia=pedro.dia;  
    juan.mes=5;  
    juan.anyo=2019;  
  
    printf("%d %d %d\n",pedro.dia,pedro.mes,pedro.anyo);  
    printf("%d %d %d\n",juan.dia,juan.mes,juan.anyo);  
}
```

6

Tablas / Arrays

7

Tablas / Arrays

- ✓ Estructura para agrupar datos del mismo tipo
- ✓ Se accede a un dato por acceso directo a través del índice de la estructura

Sintaxis

```
nombreTipo = Tabla [  $i_1, i_2$  ] de tipo_base  
nombreVariable : nombreTipo
```

Ejemplo

```
T_tempeMes = tabla [1,31] de Real  
Enero, Marzo : T_tempeMes;
```

```
Enero [1] ← -2.4  
Marzo [v] ← Marzo [v+1];
```

8

Tablas / Arrays

Otros ejemplos

Temperaturas medias año:	Temp = Tabla [1, 12] de real;
Notas en una asignatura:	Mates = Tabla [1, 40] de real;
Usuarios del bus en un mes:	BusUsers = Tabla [1, 31] de entero
Apariciones de mayúsculas	FrecuenciaLetras = Tabla ['A', 'Z'] de Entero ≥ 0
Gasto militar Europa	GastosUE = Tabla [1, 27] de Real;

9

Arrays en C/C++

```
main()
{
    int v[5];
    v[0] = 0;
    v[1] = 9;
    v[2] = 29;
    v[3] = 17;
    v[4] = 33;
}
```

Declara un array formado por 5 números enteros.

Los arrays en C siempre tienen índices desde 0 hasta n-1

$v[0], v[1], \dots, v[4]$

Mete los valores en v

V_0	0
V_1	9
V_2	29
V_3	17
V_4	33

```
#define M 3
#define N 2
main()
{
    int a[M][N];

    a[0][0] = 1; a[0][1] = 7;
    a[1][0] = 8; a[1][1] = 8;
    a[2][0] = 2; a[2][1] = 9;
}
```

Esto es un **array bidimensional** (una tabla, una matriz)

$$A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \\ a_{20} & a_{21} \end{bmatrix}$$

Se puede usar para guardar desde una matriz matemática hasta una imagen.

10

Secuencias

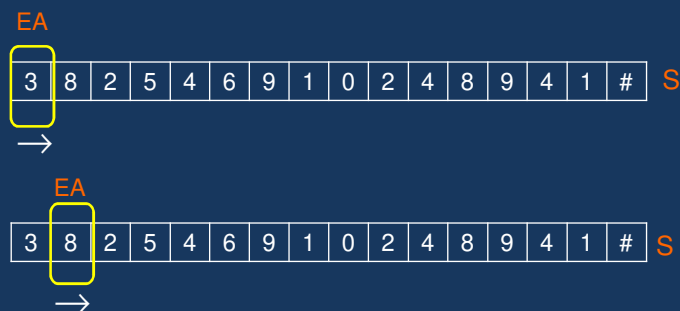
11

Secuencias (lectura)

- ✓ Conjunto de valores del mismo tipo
- ✓ Los datos se leen en orden, del primero al último dato (acceso secuencial)

Sec1 = [28, 33, 56, 1, 87, 44, 67, 12, 98, 09]
Sec2 = [V, F, F, F, V, F, F, V, V, V]

Es como tener una cinta con datos, y vamos moviendo hacia la derecha un cursor por la cinta para ir accediendo a los datos



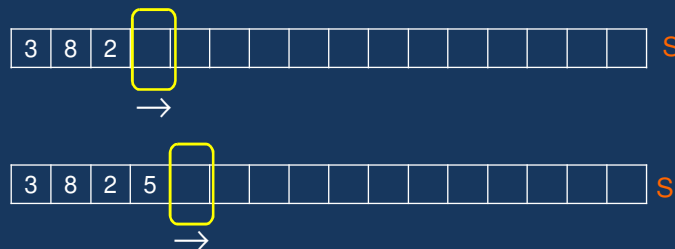
12

Secuencias (escritura)

- ✓ Conjunto de valores del mismo tipo
- ✓ Los datos se escriben en orden, añadiendo al final de la secuencia

[]
[28]
[28, 33]
[28, 33, 56]

Es como tener una cinta con datos, y vamos moviendo hacia la derecha un cursor por la cinta para ir accediendo a los datos



Secuencias

Operaciones de la máquina

De lectura:

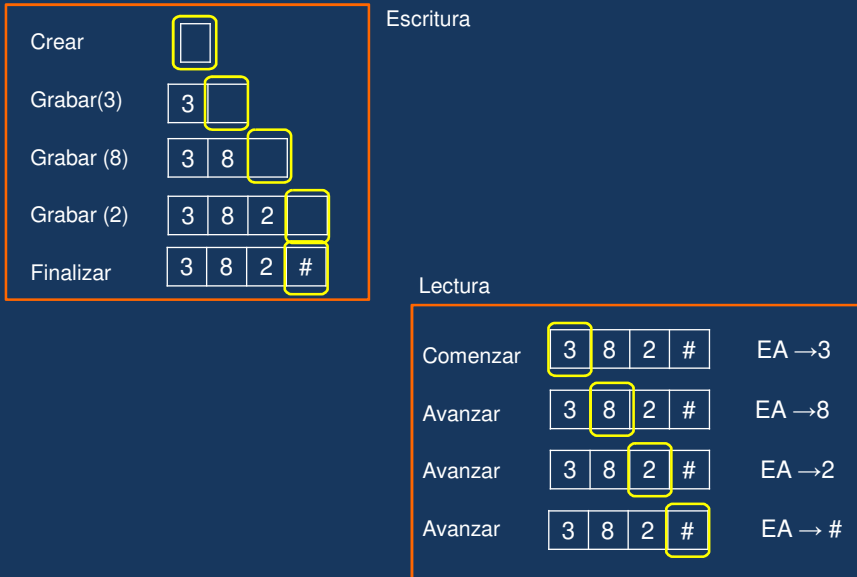
- Comenzar (S): Arranca la máquina y EA será el primer elemento de S
- Avanzar (S) : Avanza EA al siguiente
- EA(S): Extrae el elemento actual

De escritura:

- Crear(S): Crea una Secuencia vacía $S = []$
- Grabar(S,e): Concatena $S \bullet e$
- Finalizar(S): Graba # en S ($S \bullet \#$)



Secuencias



15

Implementaciones de las secuencias

- ✓ Los lenguajes pueden implementar o no estructuras de datos Secuencia
- ✓ Pero muchos problemas tienen esquema de tratamiento secuencial
- ✓ En C/C++ no existe exactamente el tipo Secuencia
- ✓ Problemas de tratamiento secuencial se tratan con acceso a archivos, lectura del teclado, gestión de listas.

16

Implementaciones de las secuencias

Con lectura de teclado

```
scanf("%d",&v);  
(tratamiento inicial)  
while (v!= Marca_Fin)  
{  
    ...(tratamiento)  
    scanf("%d",&v);  
}  
(tratamiento posterior)
```

17

Implementaciones de las secuencias

Con acceso a archivos ...

18