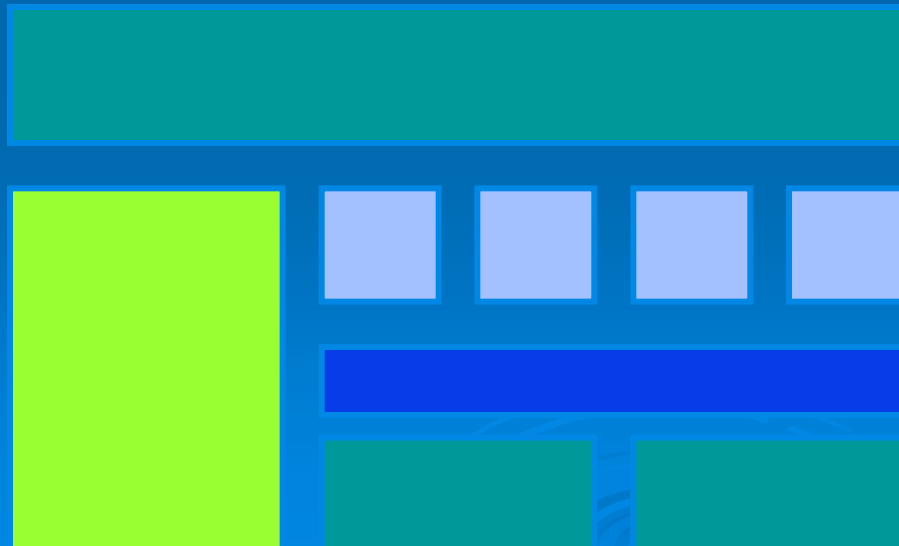


Estructuras



Introducción
a la
Programación

CONTENIDO

- Definición de estructuras. Uso de *struct*.
- Disposición de las Estructuras en memoria.
- Ejemplo de una declaración de una estructura usada en un programa.
- Creación de sinónimos o alias. Uso de *typedef*
- Arreglos de estructuras.

Definición de estructuras.

- Una estructura es un tipo de datos que permite **empaquetar** elementos bajo un **mismo nombre**. Estos elementos pueden ser de un mismo o de distinto tipos de datos, que se encuentran **relacionados** lógicamente.
- También es conocida con el nombre de **registro**.

Definición de estructuras. Uso de *struct*

En lenguaje C, se define:

```
struct Alumno{  
    char nombre[30]  
    char especialidad[30]  
    char email[50]  
};
```

→ Miembros



Definición de estructuras. Uso de *struct*

Otro ejemplo:

```
struct Alumno{  
    int dni;  
    char nombre[30];  
    char carrera[30];  
    float promedio;  
    char direccion[20];  
};
```

Una definición general de estructura

```
struct nombreEstructura
{
    TipoDato1        miembro1;
    TipoDato2        miembro2;
    .
    .
    TipoDatoN        miembroN;
}
```

Disposición de las estructuras en memoria

```
struct Datos{  
    int entero;  
    char carácter;  
    double real;  
}
```

```
sizeof(struct Datos) >= sizeof(int) +  
    sizeof(char) + sizeof(double)
```

Disposición de las estructuras en memoria



Este tipo de rellenos es necesario cuando el computador necesita que un tipo de dato real comience en una dirección de memoria que sea múltiplo de 4.

Ejemplo de una declaración de una estructura, usada en un programa.

Ejemplo1

Elabore un programa que lea y escriba los siguientes tipos de estructuras:

Un tipo estructura Complejo con dos campos reales pReal, plmaginaria.

Un tipo estructura Hora con tres campos enteros hora, minutos, segundos.

Creación de sinónimos o alias. Uso de *typedef*

- La instrucción *typedef* permite al usuario definir **alias o sinónimos**.
- El objeto de esta instrucción es utilizar **nombres más apropiados** y más cortos para los tipos de datos. **Evita** escribir la palabra *struct* en la declaración de variables.
- Ejemplo:
 - *typedef int contador;*
 - *contador c1, c2, c3;*

Creación de sinónimos o alias. Uso de *typedef*

Ejemplo 2

Elabora un programa que use una estructura que pueda describir a un restaurante; deberá tener miembros que incluyan nombres, direcciones, el precio promedio y el tipo de comida del establecimiento.

Arreglos de estructuras

- Es **frecuente** el uso conjunto de estructuras y arreglos.
- Un arreglo de estructuras representa una **lista de entidades**, que actúa como una pequeña **base de datos**, formando una tabla que tiene como identificadores de columna los **atributos** y como identificadores de fila, el **índice** del arreglo.

Arreglos de estructuras

```
#define NUMERO_FECHAS 100  
struct Fecha  
{  
    int dd;  
    int mm;  
    int aa;  
};
```

```
struct Fecha fechas[NUMERO_FECHAS];
```

Arreglos de estructuras

```
struct Fecha fechas[4]
```

dd	mm	aa
18		
		2011

fechas[0]

fechas[1]

fechas[2]

fechas[3]

fechas[3].aa = 2011

fechas[2].dd = 18