



UNIVERSIDAD
TECNOLÓGICA
DE PANAMÁ

Facultad de Ingeniería de Sistemas Computacionales

Capítulo V

Estructuras

Kexy Rodríguez

Contenido

- Conceptos Básicos.
- Declaraciones.
- Ejemplos.

Conceptos Básicos

Estructuras

En la creación de soluciones para algunos problemas surge la necesidad de agrupar datos de diferente tipo o de manejar datos que serían muy difícil de describir en los tipos de datos primitivos, esta es la situación en la que debemos aprovecharnos de las características que hacen al lenguaje C especial, o sea el uso de estructuras uniones y punteros.

Una estructura (***struct***) permite agrupar variables de varios tipos bajo un mismo nombre. A estos componentes individuales de la estructura se le denomina ***miembros***. De esta forma es posible crear elementos de datos complejos, basándose en los tipos básicos que suministra el lenguaje de programación.

Sintaxis de las estructuras

La sintaxis de la declaración de una estructura en lenguaje C es:

```
struct nombre_estructura{  
    tipo nombre_variable;  
    tipo nombre_variable;  
    ...  
    tipo nombre_variable;  
}variables_estructura;
```

Ejemplos

struct usuario

```
{  
    char nombre[25];  
    char sexo;  
    int edad;  
};
```

typedef struct

```
{  
    char nombre[25];  
    char sexo;  
    int edad;  
} usuario;
```

La palabra clave **typedef** permite solo asignarle un nuevo nombre a un tipo de datos ya existente.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
struct usuario
```

```
{
    char nombre[20];
    char apellido[20];
    char sexo;
    int edad;
};
```

```
int main() {
```

```
    //bloque de declarativa
```

```
    struct usuario cliente;
```

```
    //bloque de instrucciones
```

```
    cliente.edad=40;
```

```
    strcpy(cliente.nombre,"Pedro");
```

```
    strcpy(cliente.apellido,"Quintero");
```

```
    cliente.sexo='M';
```

```
    printf("\\n Nombre %s",cliente.nombre);
```

```
    printf("\\n Apellido %s",cliente.apellido);
```

```
    printf("\\n Edad %d",cliente.edad);
```

```
    printf("\\n Sexo %c",cliente.sexo);
```

```
    return 0;
```

```
}
```

Ejemplo N°1


```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
typedef struct
```

```
{
    char nombre[20];
    char apellido[20];
    char sexo;
    int edad;
} usuario;
```

```
int main() {
    //bloque de declarativa
    usuario cliente;
    //bloque de instrucciones
    cliente.edad=40;
    strcpy(cliente.nombre,"Pedro");
    strcpy(cliente.apellido,"Quintero");
    cliente.sexo='M';

    printf("\n Nombre %s",cliente.nombre);
    printf("\n Apellido %s",cliente.apellido);
    printf("\n Edad %d",cliente.edad);
    printf("\n Sexo %c",cliente.sexo);

    return 0;
}
```

Ejemplo N°2

Ejemplo N°3

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
typedef struct
{
    char nombre[20];
    char apellido[20];
    char sexo;
    int edad;
} usuario;

int main() {
    //bloque de declarativa
    usuario cliente[5];
    int x;
    //bloque de instrucciones
    for (x=0;x<5;x++)
    {
        printf("\nIntroduzca el nombre del cliente N°%d",x+1);
        scanf("%s",&cliente[x].nombre);
        printf("\nIntroduzca el apellido del cliente N°%d",x+1);
        scanf("%s",&cliente[x].apellido);
        printf("\nIntroduzca la edad del cliente N°%d",x+1);
        scanf("%d",&cliente[x].edad);
        printf("\nIntroduzca el sexo M/F del cliente N°%d",x+1);
        scanf("%s",&cliente[x].sexo);
    }
    system("cls");//Limpiar pantalla
    for (x=0;x<5;x++)
    {
        printf("\n ----Cliente N°%d-----",x+1);
        printf("\n Nombre %s",cliente[x].nombre);
        printf("\n Apellido %s",cliente[x].apellido);
        printf("\n Edad %d",cliente[x].edad);
        printf("\n Sexo %c",cliente[x].sexo);
    }
    return 0;
}
```


Práctica

- Crear un programa en C utilizando estructura que almacene las calificaciones de 5 estudiantes. La información del estudiante debe contener nombre, apellido, edad, sexo y calificaciones. El total de calificaciones por cada estudiante es de 5. Finalmente, el programa deberá indicar qué estudiante tiene el promedio de calificación más alta.
- Crear un programa en C utilizando estructura que almacene los datos de un grupo de 10 vehículos con las siguientes características: Marca, Modelo, Año, Color y Número de Matrícula. Finalmente, el programa deberá permitir al usuario realizar búsqueda por marca o número de matrícula, indicando si el vehículo está dentro del grupo de 10 y si está, deberá imprimir el detalle de del vehículo.



UNIVERSIDAD
TECNOLÓGICA
DE PANAMÁ

