

13-9-2021



Estructuras en c/c++

Tarea: 1

Materia: Estructura de datos

Sección: D01.

Código: 216584703

Carrera: Ingeniería en computación.

Nombre alumno: Padilla Pérez Jorge
Daray

Nombre profesor: Julio Esteban Valdes
Lopez

Índice general

Que es una Estructura en c++	2
¿Cómo se define una estructura?	2
¿Cómo se declara una variable de tipo estructura?	2
¿Qué es un campo?	3
¿Cómo se puede acceder a un campo?	3
anidación estructural	4
Acceso anidación estructural	4
Ejemplos de estructuras diferentes.....	5
Referencias bibliográficas	5

¿Qué es Una estructura en c++?

Una estructura es un grupo de variables las cuales pueden ser de diferentes tipos sostenidas o mantenidas juntas en una sola unidad. La unidad es la estructura. Las estructuras de datos se emplean con el objetivo principal de organizar los datos contenidos dentro de la memoria de la PC. Así, nuestra primera experiencia con estructuras comienza desde el momento mismo en que usamos en nuestros programas variables de tipos primitivos (char, short, int, float, etc).

¿Cómo se define una estructura?

En C/C++ se forma una estructura utilizando la palabra reservada struct, seguida por un campo etiqueta opcional, y luego una lista de miembros dentro de la estructura. La etiqueta opcional se utiliza para crear otras variables del tipo particular de la estructura:

```
struct [ <nombre tipo de estructura > ] {  
  
[<tipo> <nombre-variable[,nombre-variable,...]>];  
  
[<tipo> <nombre-variable[,nombre-variable,...]>];  
  
...  
  
} [ <variables de estructura> ] ;
```

Un punto y coma finaliza la definición de una estructura puesto que ésta es realmente una sentencia C/C++.

¿Cómo se declara una variable de tipo estructura?

Una vez que una estructura con nombre ha sido creada, la misma puede ser usada para declarar cualquier número de variables. Por ejemplo, en el siguiente fragmento de código se crea la estructura tiempo compuesta por los miembros hora, minuto y segundo; todos del tipo int. En el mismo ejemplo, se declaran las variables t1 y t2.

```
/* declaración de estructura tiempo */  
  
struct tiempo { int hora, minuto, segundo; };  
  
/* declaración de variables de tipo tiempo */  
  
struct tiempo t1, t2;
```

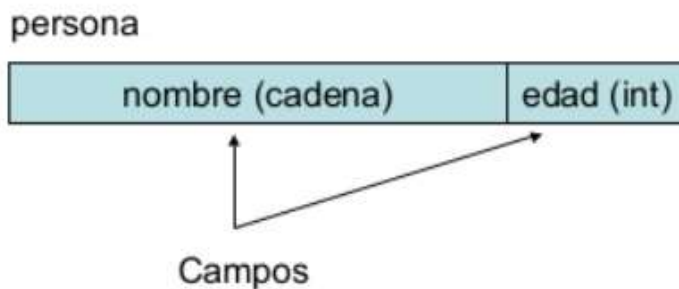
¿Qué es un campo?

Un *campo* es una variable de cualquier tipo que se declara directamente en una [clase](#) o [struct](#). Los campos son *miembros* de su tipo contenedor.

Una clase o struct puede tener campos de instancia, campos estáticos o ambos. Los campos de instancia son específicos de una instancia de un tipo. Si tiene una clase T, con un campo de instancia F, puede crear dos objetos de tipo T y modificar el valor de F en cada objeto sin afectar el valor del otro objeto. Por el contrario, un campo estático pertenece a la propia clase y se comparte entre todas las instancias de esa clase. Solo puede tener acceso al campo estático mediante el nombre de clase. Si obtiene acceso al campo estático mediante un nombre de instancia, obtendrá el error en tiempo de compilación

Cada campo esta formado por la declaración de una o más variables de algún otro tipo. Ejemplo:

```
struct persona{  
    char nombre[30];  
    int edad;  
};
```



¿Cómo se puede acceder a un campo?

Para obtener acceso a un campo en un objeto, agregue un punto después del nombre de objeto, seguido del nombre del campo, como en `objectname._fieldName`. Por ejemplo:

C#Copiar

```
CalendarEntry birthday = new CalendarEntry();
```

```
birthday.Day = "Saturday";
```

anidación estructural

Los miembros de una estructura pueden ser ellos mismos otra estructura previamente identificada o bien una estructura anónima.

Por ejemplo, en el siguiente fragmento de código se crean las estructuras pareja y pareja2. Obsérvese como dentro de los miembros de pareja2 se declara el miembro X, mismo que es una estructura del tipo pareja. Luego, las variables declaradas a raíz de la estructura pareja2 poseerán los miembros variables a y b heredados de pareja, y c.

```
struct pareja { int a, b ; };  
  
struct pareja2 { struct pareja X; int c; } P3
```

Acceso anidación estructural

Ahora bien, para acceder a los miembros de una estructura dentro de otra estructura se emplea el mismo mecanismo de acceso (el punto). Por ejemplo, para desplegar el miembro a de la variable P3 declarada en el ejemplo anterior, lo haremos más o menos así:

```
printf( "%i\n", P3.X.a );
```

Ejemplos de estructuras diferentes

Carta de baraja

numero	palo
--------	------

```
struct carta{
    int numero;
    char palo;
}
```

Libro

titulo	autor	editorial	anyo
--------	-------	-----------	------

```
struct libro{
    char titulo[40];
    char autor[30];
    char editorial[20];
    int anyo;
}
```

Polígono regular

nombre	lados	longitudLado
--------	-------	--------------

area

```
struct poligono{
    char nombre[20];
    int lados;
    float longitudLado,
        area;
}
```

```
struct alumno{
    int matricula;
    char nombre[20];
    char carrera[20];
    float promedio;
    char direccion[20];
};
```

alumno				
matricula	nombre	carrera	promedio	direccion

```
struct persona{
    char nombre[30];
    int edad;
};
```

persona

nombre (cadena)	edad (int)
-----------------	------------



Campos

Referencias bibliográficas

N.A. (2012). Estructuras c++. 13 de septiembre de 2021, de uv Sitio web: [Lenguaje de Programación: C++ Estructuras \(cimat.mx\)](#)

Bill Wagner. (20/07/2015). Campos: Guia de programacion C++. 13 de septiembre de 2021, de Microsoft Sitio web: <https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/classes-and-structs/fields#:~:text=Un%20campo%20es%20una%20variable%20de%20cualquier%20tipo,son%20espec%C3%ADficos%20de%20una%20instancia%20de%20un%20tipo.>