

Lenguaje de Programación: C++ Estructuras

José Luis Alonzo Velázquez

Universidad de Guanajuato

Octubre 2010

Declaración de una estructura

```
#include <stdio.h>
using namespace std;
//declaración con nombre
struct punto{
    int x;
    int y;
};
//declaración anónima
struct{
    int x;
    int y;
} p3;
int main(){
    return 0;
}
```

Ejemplo de uso de las estructuras descritas

```
int main(){
    punto p1,p2;
    p1.x=1;
    p1.y=1;
    p2.x=2;
    p2.y=2;
    p3.x=3;
    p3.y=3;
    printf("p1 = (%d,%d)\n",p1.x,p1.y);
    printf("p2 = (%d,%d)\n",p2.x,p2.y);
    printf("p3 = (%d,%d)\n",p3.x,p3.y);
    return 0;
}
```

Estructuras anidadas

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 cómo 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`.

Estructura por agregación o composición

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

Herencia

El termino **herencia** se usa con gran frecuencia en **Programación Orientada a Objetos**, y se le relaciona principalmente con las **clases**. Sin embargo, la herencia está presente siempre y cuando una estructura “**struct**” posea a otra estructura. En ese sentido, en C++ se presentan dos tipos de herencia:

- herencia por agregación o composición.
- herencia por extensión.

Por ejemplo, en la definición de las estructuras pareja y pareja2, se dice que pareja2 hereda por composición todos los miembros de pareja. Ahora, en el siguiente ejemplo se usa la sintaxis para que la estructura pareja2 herede por extensión los miembros de pareja:

```
// solo C++  
struct pareja { int a, b ; };  
struct pareja2 : pareja { int c; } P3;
```

Semejanza a arreglos en C

Podríamos pensar una estructura de C como una especie de arreglo que permite guardar distintos tipos de objetos en dicho arreglo.

Ejemplo

```
struct punto{
    int x;
    int y;
    int z;
};


punto p1 = { 12, 25, 3};
```

Programar una estructura triángulo

La estructura deberá guardar los tres vértices de un triángulo, además deberán calcular el centro de masas de dicho triángulo y guardarlo en una 4ta variable dentro de la estructura.

Nota

El centro de masas viene dado por el promedio de los puntos.

 Programming Principles and Practice Using C++, Bjarne Stroustrup.

 <http://www.codeblocks.org>

 <http://www.wxwidgets.org>

 (O'Reilly) Practical C Programming (3rd Edition)

 <http://www.cplusplus.com>